Nos. 23-13698-E

UNITED STATES COURT OF APPEALS
FOR THE ELEVENTH CIRCUIT

COIN CENTER, et al.,
Plaintiffs-Appellees,

v.

SECRETARY, U.S. DEPARTMENT OF THE TREASURY, et al.,
Defendants-Appellants.

Appeal from the U.S. District Court for the Northern District of Florida,
No. 3:22-cv-20375-TKW-ZCB (Wetherell, J.)

APPENDIX OF APPELLANTS COIN CENTER, ET AL.
VOL. 2 of 3

J. Abraham Sutherland
106 Connally Street
Black Mountain, NC 28711
(805) 689-4577

Jeffrey M. Harris
Cameron T. Norris
Jeffrey S. Hetzel
CONSOVOY MCCARTHY PLLC
1600 Wilson Boulevard, Suite 700
Arlington, Virginia 22209
(703) 243-9423
cam@consovoymccarthy.com

Counsel for Coin Center et al.

CASE NO. 23-13698

INDEX TO DOCUMENT REFERENCES IN APPENDIX

EXHIBIT 6

Bitcoin	Ethereum	Binance Coin	Crypto Prices	Top Assets
$21,217.95 -0.83%	$1,564.22 -3.08%	$295.47 -0.91%		

Tech

Tornado Cash Co-Founder Says the Mixer Protocol Is Unstoppable

Roman Semenov says Tornado Cash is designed so a third-party can't control it.

By Sam Reynolds Jan 25, 2022 at 3:42 a.m. EST Updated Jan 25, 2022 at 10:15 a.m. EST

Tornado Cash co-founder Roman Semenov (Roman Semenov)

The co-founder of Ethereum's best-known coin mixing service says that privacy protocols are defending people's rights to financial privacy.

Tornado Cash, a mixer that allows users to obfuscate their digital trail on the Ethereum blockchain, has been at the center of attention since it was revealed last week that hackers were using it to mix stolen ether from digital assets exchange Crypto.com.

In an interview with CoinDesk, co-founder Roman Semenov said the team has little control over what its users do with the protocol as it's designed to be autonomous and outside of the control of developers.

“There is not much we can do in terms of helping investigations because the team doesn't have much control over the protocol,” he told CoinDesk. “The Tornado Cash team mostly does research and publishes the code to GitHub. All the deployments, protocol changes and important decisions are made by the community via Tornado Governance DAO and deployment ceremonies," an event when new code is pushed live.

The way the protocol is designed, decentralized and autonomous much like decentralized finance (DeFi) protocols, means there’s nobody in charge. There’s no corporate office, executive team or CEO where the buck stops. Semenov said there’s no backend, and the user interface comes from an Ethereum Name Service domain – a service that represents Ethereum addresses as familiar-sounding domain names.

“The protocol was specifically designed this way to be unstoppable, because it wouldn't make much sense if some third party [like developers] would have control over it. This would be the same as if someone had control over Bitcoin or Ethereum,” he told CoinDesk.

Is Tornado Cash part of a criminal conspiracy?

Tornado Cash isn’t the first service to offer users the ability to mix, or tumble, their crypto. These have been around since the beginning of blockchain technology, with development efforts increasing in parallel to the ubiquity of darknet markets like Silk Road or Alpha Bay.

Law enforcement is very familiar with mixers. Bill Callahan, a retired U.S. Drug Enforcement Agency agent and now director of government affairs at the Blockchain Intelligence Group, told CoinDesk in a prior interview that he doesn’t think Tornado Cash is laundering money, equating it to running away from the police and trying to evade capture. But there would be grounds to investigate it as part of the scheme.

“If a mixer knows or maybe should have known, taken steps to know the source of the funds and the beneficial owner, and the funds are from an illicit source, they would be investigated as part of the money-laundering scheme. They could also be charged as an accessory to the crime in a criminal conspiracy,” he told CoinDesk.

In a previous statement to CoinDesk, the Financial Crimes Enforcement Network (FinCEN) said mixers like Tornado Cash may fall under the definition of a money transmitter, and therefore have “obligations” set by the Bank Secrecy Act (BSA). But it hasn’t given any further guidance.

With the high-profile takedown of darknet bitcoin mixing service Helix, then-U.S. Assistant Attorney General Brian Benczkowski said that “[obscuring] virtual currency transactions in this way is a crime.”

However, Larry Dean Harmon, the service’s operator, pled guilty and the prosecution never had to prove its case, meaning there isn’t precedent that can say with certainty that this is money laundering.

For its part, Tornado Cash’s Semenov said law enforcement hasn’t been in touch.

“Law enforcement usually knows that the developers don't have any ability to assist with an investigation or change the protocol,” he told CoinDesk.

Instead, Semenov said law enforcement would spend its time obtaining logs from infrastructure providers like Cloudflare or Infura, as these could be tied to IP addresses. Law enforcment would also likely look at any addresses linked to a centralized crypto exchange, where the wallet would have customer details linked to it via the know-your-customer (KYC) process.

“Law enforcement very rarely tries to contact us directly,” he said.

Privacy vs. security

Semenov downplayed any ideas that the protocol is a tool for criminals and said it's an important mechanism to protect the safety of crypto traders as the blockchain reveals everything for all to see.

"Since all their crypto portfolio is visible to the public, the holders of significant amounts of crypto are very vulnerable to becoming victims of kidnapping, torture and blackmail," Semenov told CoinDesk in an interview. "We think that it's a very serious threat, and the privacy protocols are very important to ensure their personal safety. The banks don't disclose your personal holdings to anyone who asks, and we think it should be the same way with crypto."

Semenov said the debate about the limits of digital privacy isn't anything new. It has always flared up any time new encryption technology has become available to retail users.

"In the 1990s, the government claimed that no strong encryption should be available to people at all, arguing that it would help terrorism," he said. "In the late 2000s, there was a similar fight over end-to-end encryption in messengers where people were defending their right to private communication."

Now, in the 2010s and 2020's, crypto is this latest frontier, and Semenov said his efforts in defending people's right to financial privacy are the "continuation of the same story that started a long time ago."

He added, “Can you imagine the world where the cypherpunks conceded from the start and we wouldn't even have HTTPS encryption of our web communications?”

Read more about

Mixers Crypto Money Laundering Security

₿	BTC	$21,217.95	▼ 0.83%	→
♦	ETH	$1,564.22	▼ 3.08%	→
◆	BNB	$295.47	▼ 0.91%	→
✕	XRP	$0.334465	▼ 2.78%	→
◆	BUSD	$1.00	▲ 0.13%	→

View All Prices

Sign up for Valid Points, our weekly newsletter breaking down Ethereum’s evolution and its impact on crypto markets.

Email address Sign Up

~~By signing up, you will receive emails about CoinDesk product updates,~~ events and marketing and you agree to our terms of services and privacy policy.

DISCLOSURE

Please note that our privacy policy, terms of use, cookies, and do not sell my personal information has been updated.

The leader in news and information on cryptocurrency, digital assets and the future of money, CoinDesk is a media outlet that strives for the highest journalistic standards and abides by a strict set of editorial policies. CoinDesk is an independent operating subsidiary of Digital Currency Group, which invests in cryptocurrencies and blockchain startups. As part of their compensation, certain CoinDesk employees, including editorial employees, may receive exposure to DCG equity in the form of stock appreciation rights, which vest over a multi-year period. CoinDesk journalists are not allowed to purchase stock outright in DCG.

Sam Reynolds

Follow @thesamreynolds on Twitter

Trending

1 Markets

First Mover Americas: Macro Retakes Front Seat, Pushing Bitcoin Down Below $21K

Aug 22, 2022

2 Sponsored Content

Creating a Sustainable GameFi and NFT Landscape for Both Players and Projects

Aug 22, 2022

3 Business

Wall Street's DTCC Launches Private Blockchain Platform to Settle Trades

Aug 22, 2022

4 Tech

Popular Uniswap Frontend Blocks Over 250 Crypto Addresses Related to DeFi Crimes

Aug 22, 2022

About

About
Masthead
Contributors
Careers
Company News

Get In Touch

Contact Us
Advertise
Accessibility Help
Sitemap

Stay Updated

Events
Newsletters
Follow

The Fine Print

Ethics Policy
Privacy
Terms Of Use
Do Not Sell My Personal Information

Please note that our privacy policy, terms of use, cookies, and do not sell my personal information has been updated.

The leader in news and information on cryptocurrency, digital assets and the future of money, CoinDesk is a media outlet that strives for the highest journalistic standards and abides by a strict set of editorial policies. CoinDesk is an independent operating subsidiary of Digital Currency Group, which invests in cryptocurrencies and blockchain startups. As part of their compensation, certain CoinDesk employees, including editorial employees, may receive exposure to DCG equity in the form of stock appreciation rights, which vest over a multi-year period. CoinDesk journalists are not allowed to purchase stock outright in DCG.

English

EXHIBIT 7.

Bitcoin BTC (24h) **$21,336.52** +0.56%

Ethereum ETH (24h) **$1,580.96** -0.28%

Binance Coin BNB (24h)

View All

Price data by CoinMarketCap

▸ **News** ▸ **Coins**

Tornado Cash Ethereum Token TORN Down 50% After Sanctions

The coin mixer's governance token price has steadily declined since regulators took action on Monday.

By Jason Nelson

Aug 12, 2022

2 min read

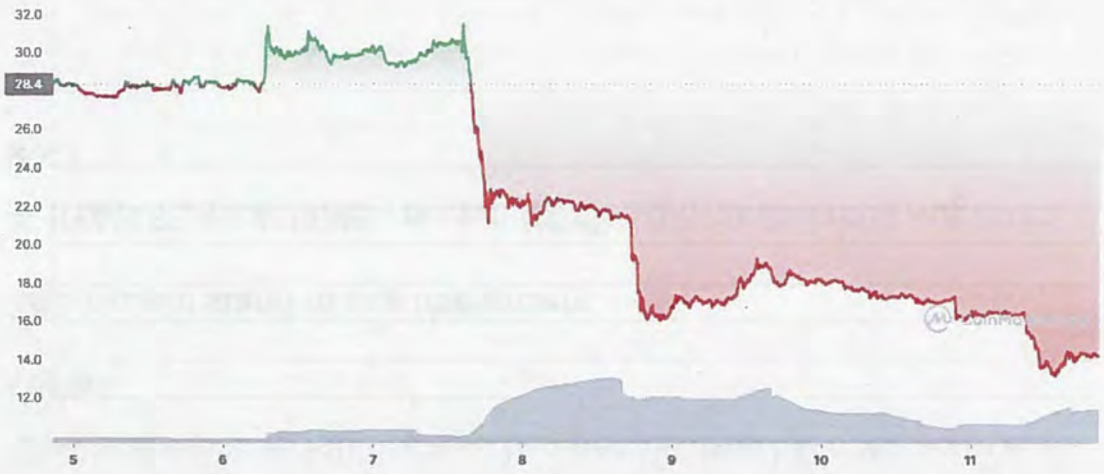
Image: Shutterstock

In the week following the blacklisting of Tornado Cash by the

Tornado Cash Ethereum Token TORN Down 50% After Sanctions

56%, starting the week at a high of $31.56 and ending the week at a low of $13.09, according to CoinMarketCap.

Tornado Cash (TORN) is an ERC20 token and the native token of the Tornado Cash DAO, used to manage governance and voting. It's currently the No. 691 cryptocurrency, with a market capitalization of $15.5 million.



After the U.S. Treasury issued its sanctions and Github took the Tornado Cash website offline by removing its repository from the site, the token price began to slide.

Launched in 2019, Tornado Cash is a blockchain protocol for sending and receiving anonymous transactions by mixing Ethereum tokens with a pool of other tokens, anonymizing the user.

In its sanctioning of Tornado Cash, the U.S. Treasury cited its use by the North Korean hacker group Lazarus Group and the laundering of over $103.8 million from the hacks of the Horizon

Harmony Bridge and Nomad Token Bridge earlier this summer.

Following a debate by the Tornado Cash community, the Discord server for the group disappeared, and unknown persons took the forum on the Tornado Cash community website offline as well. At the same time, a member of the developer group behind Tornado Cash was taken into custody by law enforcement in the Netherlands.

> They arrested the developer of tornado cash. 🚨
>
> I repeat: a man was arrested for writing code that served as a public good for people to maintain their privacy online.
>
> They put a man in jail because bad people used his open source code.
>
> This cannot stand in any free society.
>
> — RYAN SΞAN ADAMS - rsa.eth 🦇🔊 (@RyanSAdams) August 12, 2022

The Fiscal Information and Investigation Service (FIOD) said its criminal investigation into Tornado Cash began in June 2022 criminal investigation into Tornado Cash began in June 2022.

This crypto winter appears to be especially harsh for the Tornado Cash community. Coupled with the current bear market, sanctions, shutdowns, and arrests seem to have dealt a body blow to the project as holders continue to flee.

Stay on top of crypto news, get daily updates in your inbox.

Your Email

you@emailaddress.com

Get it!

Copy link

IPFS

Recommended News

This Week on Crypto Twitter: Tornado Cash Crackdown Crashes Crypto Party, North Korean Hackers Post Fake Coinbase Jobs

Illustration by Mitchell Preffer for Decrypt With crypto prices cooling by double-digit percentages over the week as Autumn approaches, it's fair to say that any respite from crypto's ongoing ...

News ‣ Coins

4 min read

Tim Hakki Aug 21, 2022

This Week in Coins: Bitcoin and Ethereum Drop by Double Digits

This week in coins. Illustration by Mitchell Preffer for Decrypt. It was a very red week for crypto, with most leading cryptocurrencies depreciating over the seven days as heavyweights like B...

News ‣ Coins

5 min read

Tim Hakki Aug 20, 2022

Ethereum Privacy Platform Aztec Responds to FTX 'Freezing' User Accounts

Aztec Network, a privacy-enhancing smart contract platform on Ethereum, took to Twitter on Friday to respond to customer complaints that FTX, one of the largest cryptocurrency exchanges in the...

News ‣ Coins 3 min read

Jason Nelson Aug 19, 2022

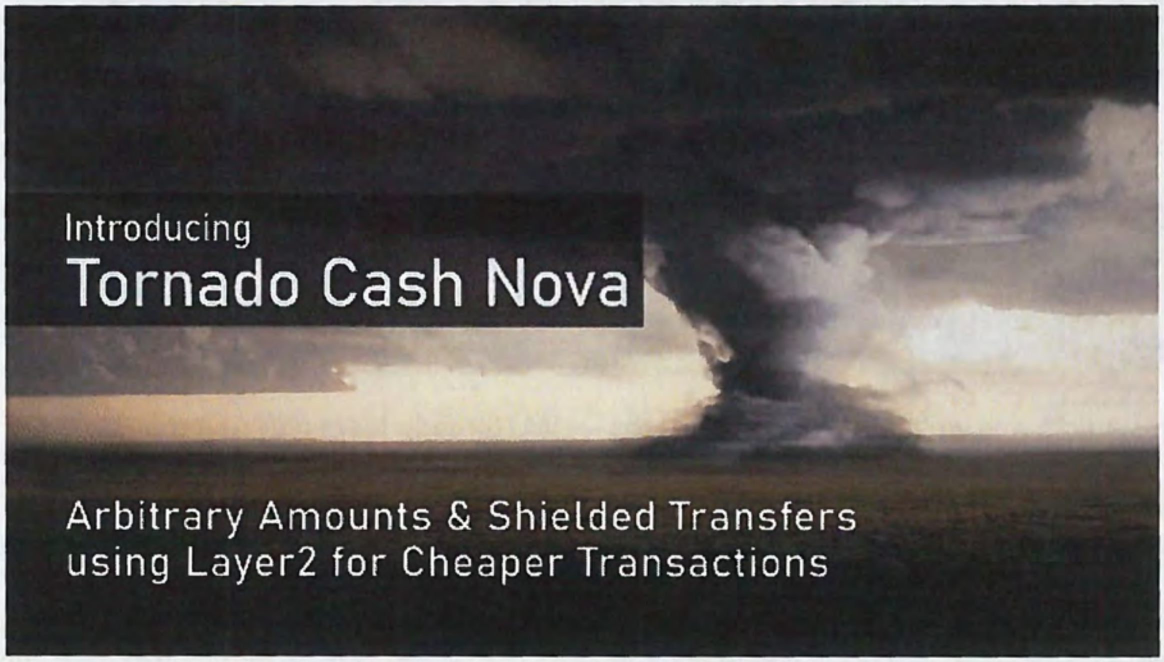
All News ➞

EXHIBIT 15

Tornado Cash Introduces Arbitrary Amounts & Shielded Transfers

tornado-cash.medium.com/tornado-cash-introduces-arbitrary-amounts-shielded-transfers-8df92d93c37c

Tornado Cash December 15, 2021


Introducing
Tornado Cash Nova
Arbitrary Amounts & Shielded Transfers
using Layer2 for Cheaper Transactions

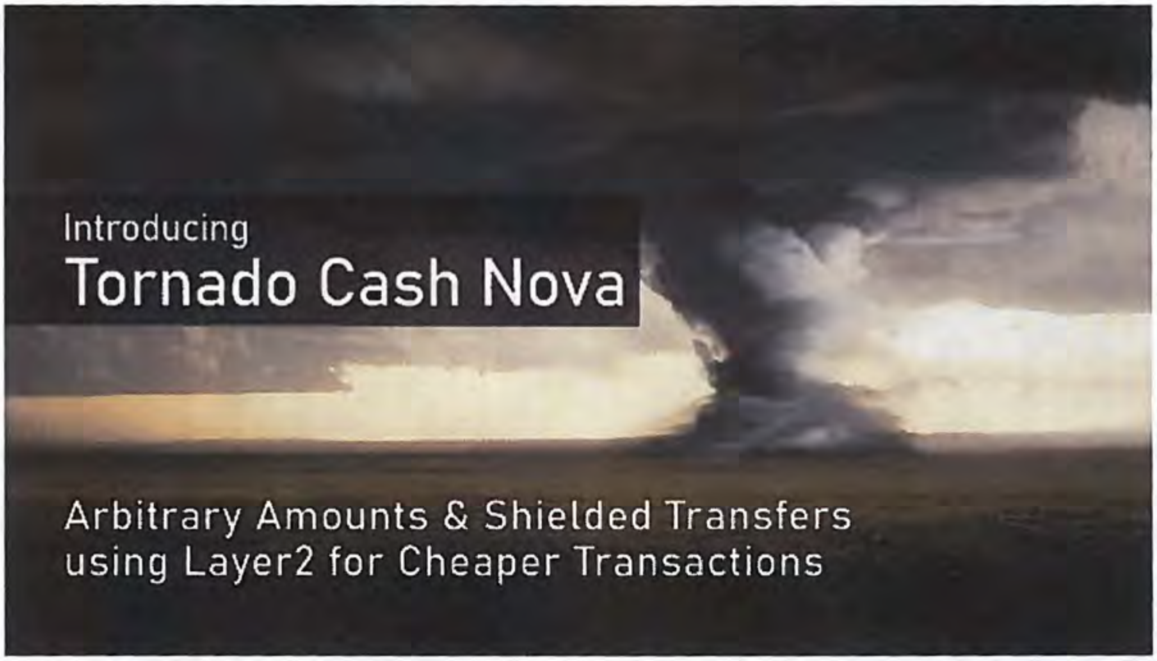
Tornado Cash

Dec 15, 2021

.

5 min read

What if we told you that fixed amounts pools may no longer be your sole option on Tornado Cash? , the beta version of a new ETH pool allowing both arbitrary amounts & shielded transfers, was recently deployed and is ready for you to dive in.


Introducing
Tornado Cash Nova
Arbitrary Amounts & Shielded Transfers
using Layer2 for Cheaper Transactions

© This article was written by ayefda & bt11ba — WUTornado team (@WUTornado)

We are glad to introduce , an upgraded Tornado Cash pool presenting unique features focused on improving user experience & expanding the protocol functionalities.

This pool will allow users to deposit & withdraw arbitrary amounts of ETH. Up to now, all Tornado Cash pools had one thing in common: users could only deposit and withdraw a fixed amount of a given token within each pool. With the arrival of the Nova pool, this statement will no longer be true.

This new version will also provide the possibility to make shielded transfers of deposited tokens while staying within the pool. So far, to transfer the custody of deposited funds, tokens needed to be withdrawn first. That is about to change! Users will be able to transfer a chosen amount of their deposited tokens (not necessarily all of them) to another address without needing to withdraw them from the pool.

Speed & cost being the cornerstone of user experience, Tornado Cash Nova uses the Gnosis Chain as a Layer2 (former xDAI). Thanks to this, users can benefit from cheaper fees, while still having fast transactions.
Indeed, this sidechain was specifically chosen as an L2 for being the only one that supports fast withdrawals to Mainnet *(a few minutes vs. ~3hours on Polygon &~7 days on Optimism or Arbitrum)*.

⚙ Let us take a peek at how Tornado Cash Nova functions.

First step: Diving into the pool 💰

A “Fund” section will allow you to deposit ETH within this new pool.

The great novelty is that deposited amounts will no longer be predefined. You can choose the customized amount that you want to deposit.Therefore, if you want to put 0.4 ETH in the pool, you can do it all at once rather than making four separate transactions in the 0.1 ETH pool.

⚠️ Since it is a beta version, **deposits will be limited at 1 ETH/transaction**. However, if the community wishes to increase this limit, the 1 ETH cap amount can always be changed through a governance proposal.

As soon as your wallet is connected to the DApp,the wallet address will be considered the default "Shielded address" of the funds. The shielded address is the one that will allow you to log into the app later, check your Tornado shielded balance or receive shielded transfers. Each new address will be registered to the pool alongside its first deposit.

However,the shielded address doesn't necessarily need to be your connected wallet address. You can use your wallet address to deposit tokens to any shielded address of your choice (yours or anyone else's) as long as this address is registered to Tornado Cash Nova.

Indeed, depositing funds isn't the sole option to register a shielded address to the pool. Users can also add new shielded addresses as a separate action through the DApp.

Intermediary step: Sending your tokens away 💸

A "Transfer" section will allow shielded transactions within the pool.

As announced above, tokens can be transferred from one address to another within Tornado Cash Nova. The transferred amount can be completely customized. Those shielded transfers can be sent to any chosen address as long as this address is also registered in Tornado Cash Nova.

As a reminder, a shielded address is registered either completely ad hoc through the DApp or alongside an initial deposit from a wallet address.

⚠️ Two transfer methods are available to provide ETH in order to pay the gas fee for a transaction. You can either connect your wallet or use a Relayer.
Going through your wallet for this gas fee can compromise the anonymity of the transfer if used ETH is linkable to your identity. Therefore, it is recommended to use a Relayer to preserve privacy.

Last step: Leaving the pool anonymously 👋

The last but not least — a **"Withdraw" section** to pull your funds out of the pool.

To withdraw your funds from the Tornado pool, you can either choose from a set of four predetermined amounts (0.1, 0.3, 0.5 & 1 ETH) or a completely customized amount.

Choosing one of these four amounts is strongly recommended as it will allow your withdrawal to blend with the crowd and remain anonymous. With a custom amount, a link might be deduced between your initial fund transaction and your withdrawal, which might compromise your privacy.

Besides, a connection between deposits & withdrawals might also be made if the initially funded amount & the withdrawn amount are exactly the same or easily calculated through a sum. Indeed, a deposit of 0.42 ETH can be linked to a withdrawal of exactly 0.42 ETH or two times 0.21 ETH, which might compromise anonymity. However, with a withdrawal of 0.391 ETH, privacy is better preserved as there is no obvious link between the 0.42 & 0.391 amounts.

Therefore, the custom option should only be chosen with full knowledge of these facts and in complete confidence in your actions.

To obtain full privacy, the same good practices that were recommended for traditional Tornado Cash pools are still required. You can find a few tips to remain anonymous in the protocol's documentation.

For cheaper transactions, Gnosis Chain (xDAI) is used as a Layer-2. To this end, a bridge is used between ETH from the Mainnet & WETH from Gnosis Chain.
Therefore, to prevent spam attacks that will overload the bridge, the withdrawal amount has to be larger than 0.05 ETH.

⚠️ As for shielded transfers, two withdrawal methods are available to pay for the gas fee. You can either connect your wallet or use a Relayer.
To preserve privacy, the use of a Relayer is also recommended for withdrawal. Paying the gas fee from your wallet might compromise your anonymity by linking your withdrawal to the used wallet address.

Few other facts

- As mentioned in the withdrawal section, Tornado Cash Nova will be using Gnosis chain (former xDAI) as a Layer-2 so that all related transactions are cheaper. The use of this L2 is possible thanks to the ETH <> WETH bridge available on xDAI.
- Once deployed, all those contracts can be upgraded by the governance. Indeed, the xDAI bridge supports the transfer of messages between the L1 & the L2.
- As privacy still rhymes with accountability, a compliance tool will soon be available for all transactions made within this pool.
- The code was audited by Zeropool.

Future Plans

An improved V3 of Tornado Cash is being currently prepared. This incoming version mainly focuses on enhancing users' experience. Some handy new features are planned to bring more flexibility & possibilities to the use of the protocol.

With its customized amounts & shielded transfers, Tornado Cash Nova is the first step towards this new & improved version of Tornado Cash. Future plans for the protocol include the possibility of making atomic swaps within a shielded pool, as well as a pool that will be able to support ERC20 tokens & NFT.

Try it now at

This article was written by & — team ()

EXHIBIT 54

U.S. DEPARTMENT OF THE TREASURY

Treasury Takes Robust Actions to Counter Ransomware

September 21, 2021

Targets First Virtual Currency Exchange for Laundering Cyber Ransoms

OFAC Updates Ransomware Advisory to Encourage Reporting and Cyber Resilience

WASHINGTON — As part of the whole-of-government effort to counter ransomware, the U.S. Department of the Treasury today announced a set of actions focused on disrupting criminal networks and virtual currency exchanges responsible for laundering ransoms, encouraging improved cyber security across the private sector, and increasing incident and ransomware payment reporting to U.S. government agencies, including both Treasury and law enforcement. Treasury's actions today advance the United States government's broader counter-ransomware strategy, which emphasizes the need for a collaborative approach to counter ransomware attacks, including partnership between the public and private sector and close relationships with international partners.

"Ransomware and cyber-attacks are victimizing businesses large and small across America and are a direct threat to our economy. We will continue to crack down on malicious actors," said Treasury Secretary Janet L. Yellen. "As cyber criminals use increasingly sophisticated methods and technology, we are committed to using the full range of measures, to include sanctions and regulatory tools, to disrupt, deter, and prevent ransomware attacks."

Ransomware attacks are increasing in scale, sophistication, and frequency, victimizing governments, individuals, and private companies around the world. In 2020, ransomware payments reached over $400 million, more than four times their level in 2019. The U.S. government estimates that these payments represent just a fraction of the economic harm caused by cyber-attacks, but they underscore the objectives of those who seek to weaponize technology for personal gain: to disrupt our economy and damage the companies, families, and individuals who depend on it for their livelihoods, savings, and futures. In addition to the millions of dollars paid in ransoms and recovery, the disruption to critical sectors, including financial services, healthcare, and energy, as well as the exposure of confidential information, can cause severe damage.

Some virtual currency exchanges are a critical element of this ecosystem, as virtual currency is the principal means of facilitating ransomware payments and associated money laundering activities. The United States has been a leader in applying its anti-money laundering/countering the financing of terrorism (AML/CFT) framework in the virtual currency area, including with the Financial Crimes Enforcement Network (FinCEN) publishing guidance regarding the application of Bank Secrecy Act rules in this area in 2013 and 2019. FinCEN has also taken important enforcement action against non-compliant virtual currency money transmitters facilitating ransomware payments, such as BTC-e in 2017 and the virtual currency mixing service Helix in 2020. In addition, the United States is taking steps to improve transparency regarding ransomware attacks and associated payments.

DESIGNATION OF FIRST VIRTUAL CURRENCY EXCHANGE FOR COMPLICIT FINANCIAL SERVICES

Today's actions include the Department of the Treasury's Office of Foreign Assets Control's (OFAC) designation of SUEX OTC, S.R.O. (SUEX), a virtual currency exchange, for its part in facilitating financial transactions for ransomware actors. SUEX has facilitated transactions involving illicit proceeds from at least eight ransomware variants. Analysis of known SUEX transactions shows that over 40% of SUEX's known transaction history is associated with illicit actors. SUEX is being designated pursuant to Executive Order 13694, as amended, for providing material support to the threat posed by criminal ransomware actors.

Virtual currency exchanges such as SUEX are critical to the profitability of ransomware attacks, which help fund additional cybercriminal activity. Treasury will continue to disrupt and hold accountable these entities to reduce the incentive for cybercriminals to continue to conduct these attacks. This action is the first sanctions designation against a virtual currency exchange and was executed with assistance from the Federal Bureau of Investigation.

While most virtual currency activity is licit, virtual currencies can be used for illicit activity through peer-to-peer exchangers, mixers, and exchanges. This includes the facilitation of sanctions evasion, ransomware schemes, and other cybercrimes. Some virtual currency exchanges are exploited by malicious actors, but others, as is the case with SUEX, facilitate illicit activities for their own illicit gains. Treasury will continue to use its authorities against malicious cyber actors in concert with other U.S. departments and agencies, as well as our foreign partners, to disrupt financial nodes tied to ransomware payments and cyber-attacks. Those in the virtual currency industry play a critical role in implementing appropriate AML/CFT and sanctions controls to prevent sanctioned persons and other illicit actors from exploiting virtual currencies to undermine U.S foreign policy and national security interests.

SANCTIONS IMPLICATIONS

As a result of today's designation, all property and interests in property of the designated target that are subject to U.S. jurisdiction are blocked, and U.S. persons are generally prohibited from engaging in transactions with them. Additionally, any entities 50% or more owned by one or more designated persons are also blocked. In addition, financial institutions and other persons that engage in certain transactions or activities with the sanctioned entities and individuals may expose themselves to sanctions or be subject to an enforcement action. Today's action against SUEX does not implicate a sanctions nexus to any particular Ransomware-as-a-Service (RaaS) or variant.

OFAC UPDATES ADVISORY ON POTENTIAL SANCTIONS RISKS FOR FACILITATING RANSOMWARE PAYMENTS

OFAC today also released an Updated Advisory on Potential Sanctions Risks for Facilitating Ransomware Payments . The Advisory emphasizes that the U.S. government continues to strongly discourage the payment of cyber ransom or extortion demands and recognizes the importance of cyber hygiene in preventing or mitigating such attacks. OFAC has also updated the Advisory to emphasize the importance of improving cybersecurity practices and reporting to, and cooperating with, appropriate U.S. government agencies in the event of a ransomware attack. Such reporting, as the Advisory notes, is essential for U.S. government agencies, including law enforcement, to understand and counter ransomware attacks and malicious cyber actors.

OFAC strongly encourages victims and related companies to report these incidents to and fully cooperate with law enforcement as soon as possible to avail themselves of OFAC's significant mitigation related to OFAC enforcement matters and receive voluntary self-disclosure credit in the event a sanctions nexus is later determined.

ADDITIONAL AUTHORITIES

FinCEN, in addition to the guidance and enforcement activities above, has also engaged with industry, law enforcement, and others on the ransomware threat through the FinCEN Exchange public-private partnership. FinCEN held a first Exchange on ransomware in November 2020 and a second Exchange in August 2021. FinCEN is taking additional action under its authorities to collect information relating to ransomware payments.

INTERNATIONAL COOPERATION AND IMPORTANCE OF AML/CFT MEASURES FOR VIRTUAL CURRENCIES AND SERVICE PROVIDERS

Countering ransomware benefits from close collaboration with international partners. At the Group of Seven (G7) meeting in June, participants committed to working together to urgently address the escalating shared threat from criminal ransomware networks. The G7 is considering the risks surrounding ransomware, including potential impacts to the finance sector. For example, the G7 Cyber Expert Group (CEG), co-chaired by Treasury and Bank of England, met on September 1 and September 14, 2021 to discuss ransomware, which remains a grave concern given the number and breadth of ransomware attacks across industry sectors. The participants considered the effects of ransomware attacks on the financial services sector, as well as the broader economy, and explored ways to help improve overall security and resilience against malicious cyber activity.

Given the illicit finance risk that virtual assets pose, including ransomware-related money laundering, in June 2019 the Financial Action Task Force (FATF) amended its standards to require all countries to regulate and supervise virtual asset service providers (VASPs), including exchanges, and to mitigate against such risks when engaging in virtual asset transactions. Among other things, countries are expected to impose customer due diligence (CDD) requirements, and suspicious transaction reporting obligations across VASPs, which can help inhibit cybercriminals' exploitation of virtual assets while supporting investigations into these illicit finance activities. Because profit-motivated cybercriminals must launder their misappropriated funds, AML/CFT regimens are a critical chokepoint in countering and deterring this criminal activity. This magnifies the need for all countries to effectively and expeditiously implement and enforce the FATF's standards on virtual assets and VASPs. The United States is committed to continued work at the FATF and with other countries to implement the FATF standards, and we welcome the FATF's ongoing work on this issue.

Click here to view identifying information on the entity designated today.

Click here for OFAC's Frequently Asked Questions on Virtual Currency.

FOR MORE INFORMATION ON RANSOMWARE

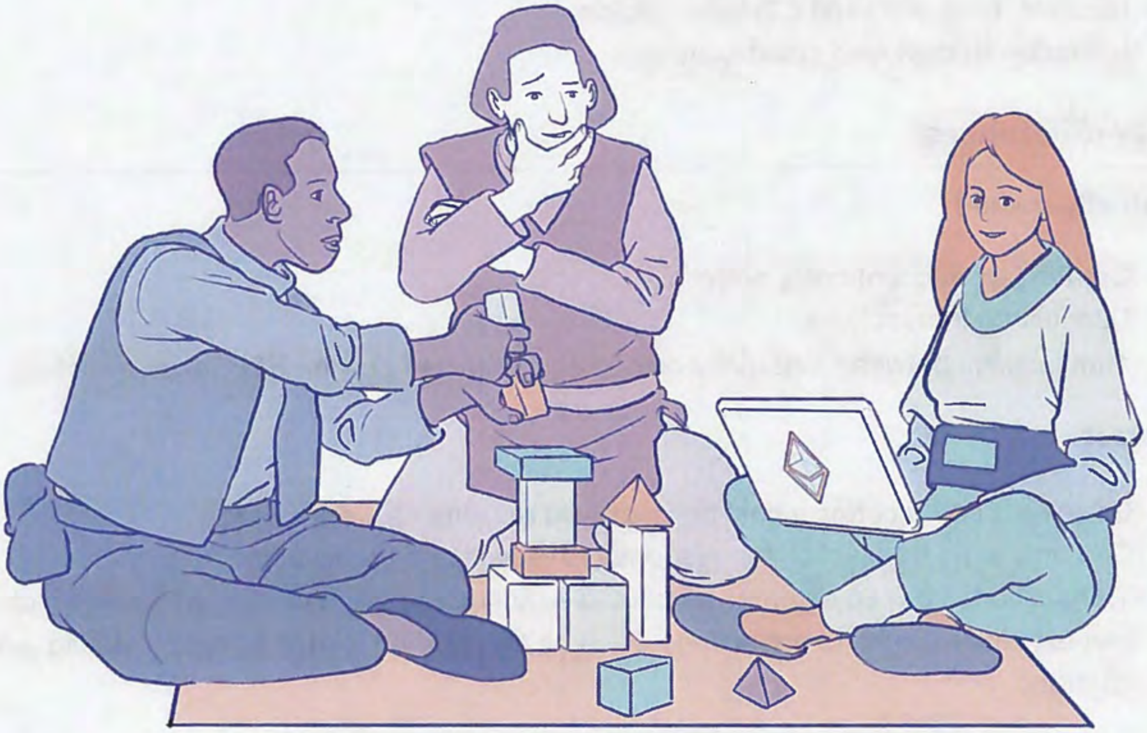
Please visit StopRansomware.gov, a one-stop resource for individuals and organizations of all sizes to reduce their risk of ransomware attacks and improve their cybersecurity resilience. This webpage brings together tools and resources from multiple federal government agencies under one online platform. Learn more about how ransomware works, how to protect yourself, how to report an incident, and how to request technical assistance.

###

EXHIBIT 58

Ethereum accounts

ethereum.org/en/developers/docs/accounts/



Last edit: , Invalid DateTime

Edit page

On this page

An Ethereum account is an entity with an ether (ETH) balance that can send transactions on Ethereum. Accounts can be user-controlled or deployed as smart contracts.

Prerequisites

Accounts are a very beginner-friendly topic. But to help you better understand this page, we recommend you first read through our introduction to Ethereum.

Account types

Ethereum has two account types:

- Externally-owned – controlled by anyone with the private keys
- Contract – a smart contract deployed to the network, controlled by code. Learn about smart contracts

Both account types have the ability to:

- Receive, hold and send ETH and tokens
- Interact with deployed smart contracts

Key differences

Externally-owned

- Creating an account costs nothing
- Can initiate transactions
- Transactions between externally-owned accounts can only be ETH/token transfers

Contract

- Creating a contract has a cost because you're using network storage
- Can only send transactions in response to receiving a transaction
- Transactions from an external account to a contract account can trigger code which can execute many different actions, such as transferring tokens or even creating a new contract

An account examined

Ethereum accounts have four fields:

- `nonce` – A counter that indicates the number of transactions sent from the account. This ensures transactions are only processed once. In a contract account, this number represents the number of contracts created by the account.
- `balance` – The number of wei owned by this address. Wei is a denomination of ETH and there are 1e+18 wei per ETH.
- `codeHash` – This hash refers to the *code* of an account on the Ethereum virtual machine (EVM). Contract accounts have code fragments programmed in that can perform different operations. This EVM code gets executed if the account gets a message call. It cannot be changed, unlike the other account fields. All such code fragments are contained in the state database under their corresponding hashes for later retrieval. This hash value is known as a codeHash. For externally owned accounts, the codeHash field is the hash of an empty string.

- `storageRoot` – Sometimes known as a storage hash. A 256-bit hash of the root node of a Merkle Patricia trie that encodes the storage contents of the account (a mapping between 256-bit integer values), encoded into the trie as a mapping from the Keccak 256-bit hash of the 256-bit integer keys to the RLP-encoded 256-bit integer values. This trie encodes the hash of the storage contents of this account, and is empty by default.

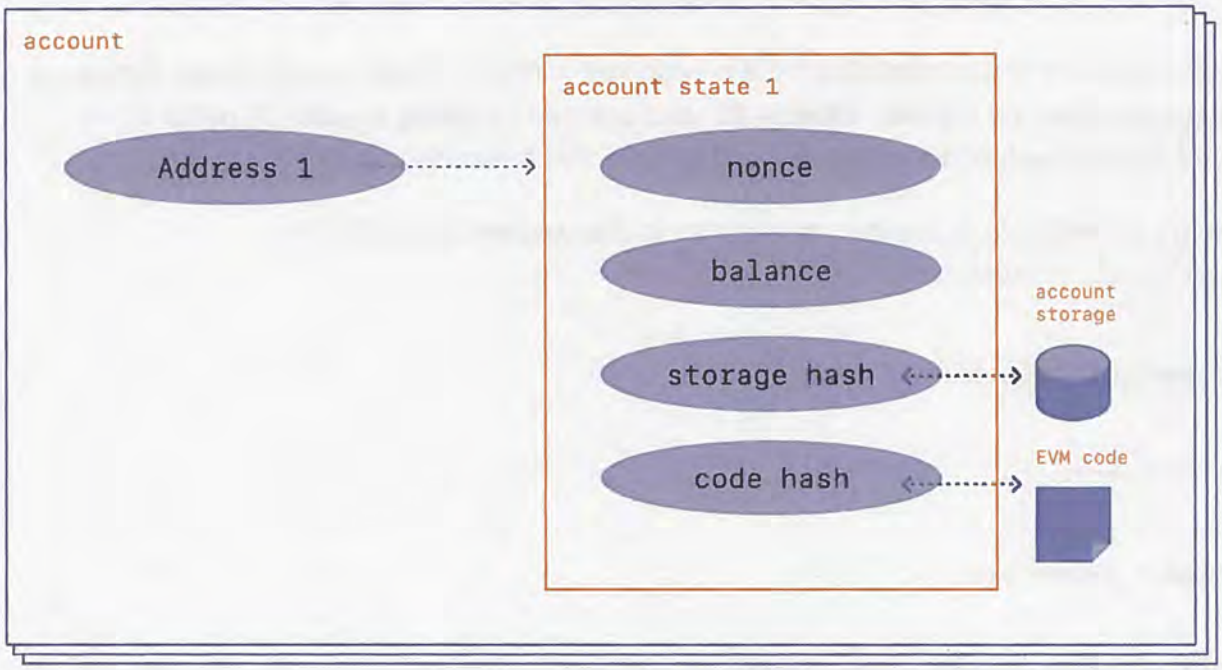


Diagram adapted from Ethereum EVM illustrated

Externally-owned accounts and key pairs

An account is made up of a cryptographic pair of keys: public and private. They help prove that a transaction was actually signed by the sender and prevent forgeries. Your private key is what you use to sign transactions, so it grants you custody over the funds associated with your account. You never really hold cryptocurrency, you hold private keys – the funds are always on Ethereum's ledger.

This prevents malicious actors from broadcasting fake transactions because you can always verify the sender of a transaction.

If Alice wants to send ether from her own account to Bob's account, Alice needs to create a transaction request and send it out to the network for verification. Ethereum's usage of public-key cryptography ensures that Alice can prove that she originally initiated the transaction request. Without cryptographic mechanisms, a malicious adversary Eve could simply publicly broadcast a request that looks something like "send 5 ETH from Alice's account to Eve's account," and no one would be able to verify that it didn't come from Alice.

Account creation

When you want to create an account most libraries will generate you a random private key.

A private key is made up of 64 hex characters and can be encrypted with a password.

Example:

```
fffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd036415f
```

The public key is generated from the private key using the Elliptic Curve Digital Signature Algorithm. You get a public address for your account by taking the last 20 bytes of the Keccak-256 hash of the public key and adding `0x` to the beginning.

Here's an example of creating an account in the console using GETH's `personal_newAccount`

```
> personal.newAccount()

Passphrase:

Repeat passphrase:

"0x5e97870f263700f46aa00d967821199b9bc5a120"



> personal.newAccount("h4ck3r")

"0x3d80b31a78c30fc628f20b2c89d7ddbf6e53cedc"

```

GETH documentation

It is possible to derive new public keys from your private key but you cannot derive a private key from public keys. This means it's vital to keep a private key safe and, as the name suggests, **PRIVATE**.

You need a private key to sign messages and transactions which output a signature. Others can then take the signature to derive your public key, proving the author of the message. In your application, you can use a javascript library to send transactions to the network.

Contract accounts

Contract accounts also have a 42 character hexadecimal address:

Example:

```
0x06012c8cf97bead5deae237070f9587f8e7a266d
```

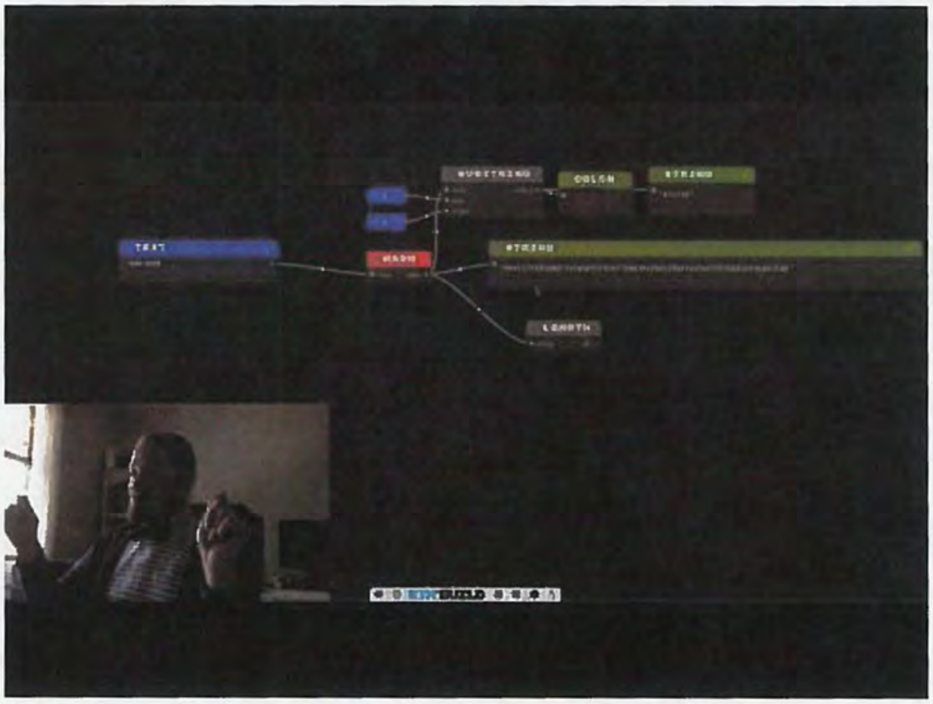
The contract address is usually given when a contract is deployed to the Ethereum Blockchain. The address comes from the creator's address and the number of transactions sent from that address (the “nonce”).

A note on wallets

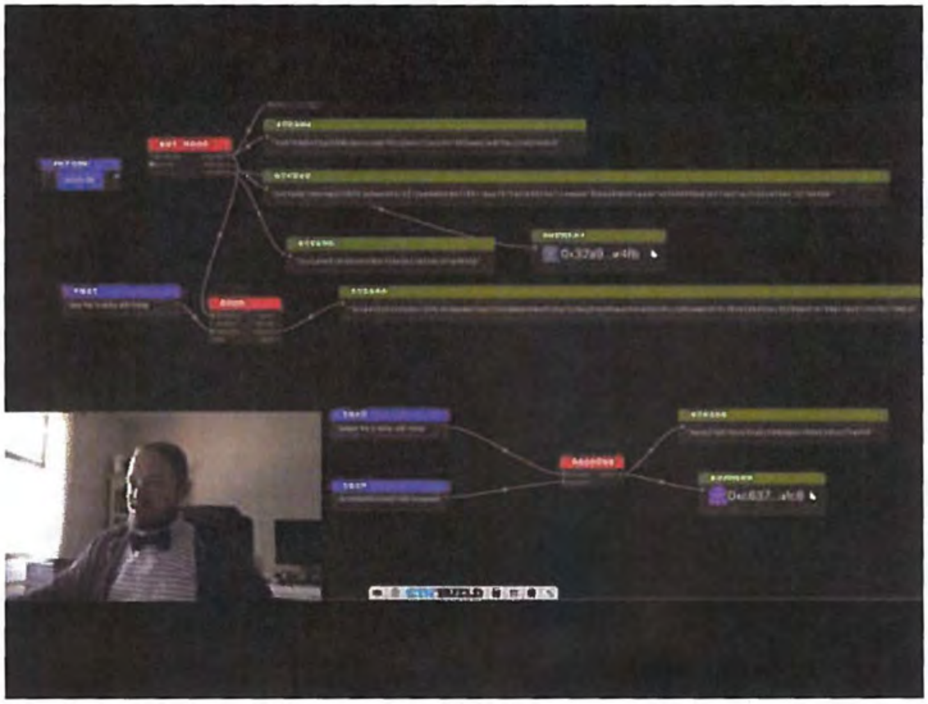
An account is not a wallet. An account is the keypair for a user-owned Ethereum account. A wallet is an interface or application that lets you interact with your Ethereum account.

A visual demo

Watch Austin walk you through hash functions, and key pairs.



Watch Video At: https://youtu.be/QJ010l-pBpE



Watch Video At: https://youtu.be/9LtBDy67Tho

Further reading

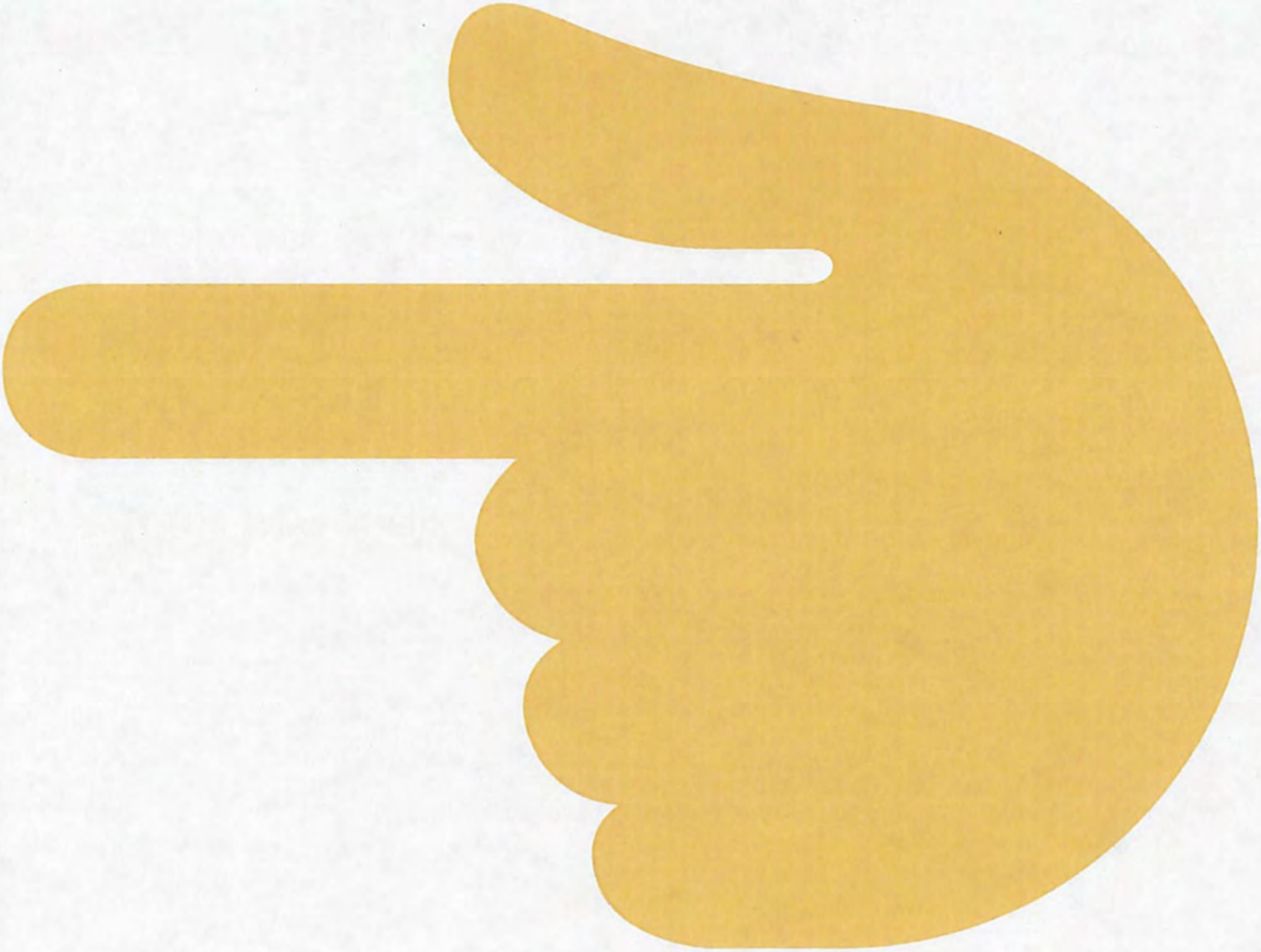
Know of a community resource that helped you? Edit this page and add it!

Related topics

- Smart contracts
- Transactions

Back to top ↑

Was this article helpful?



Previous<u>Web2 vs Web3</u>
Next<u>Transactions</u>

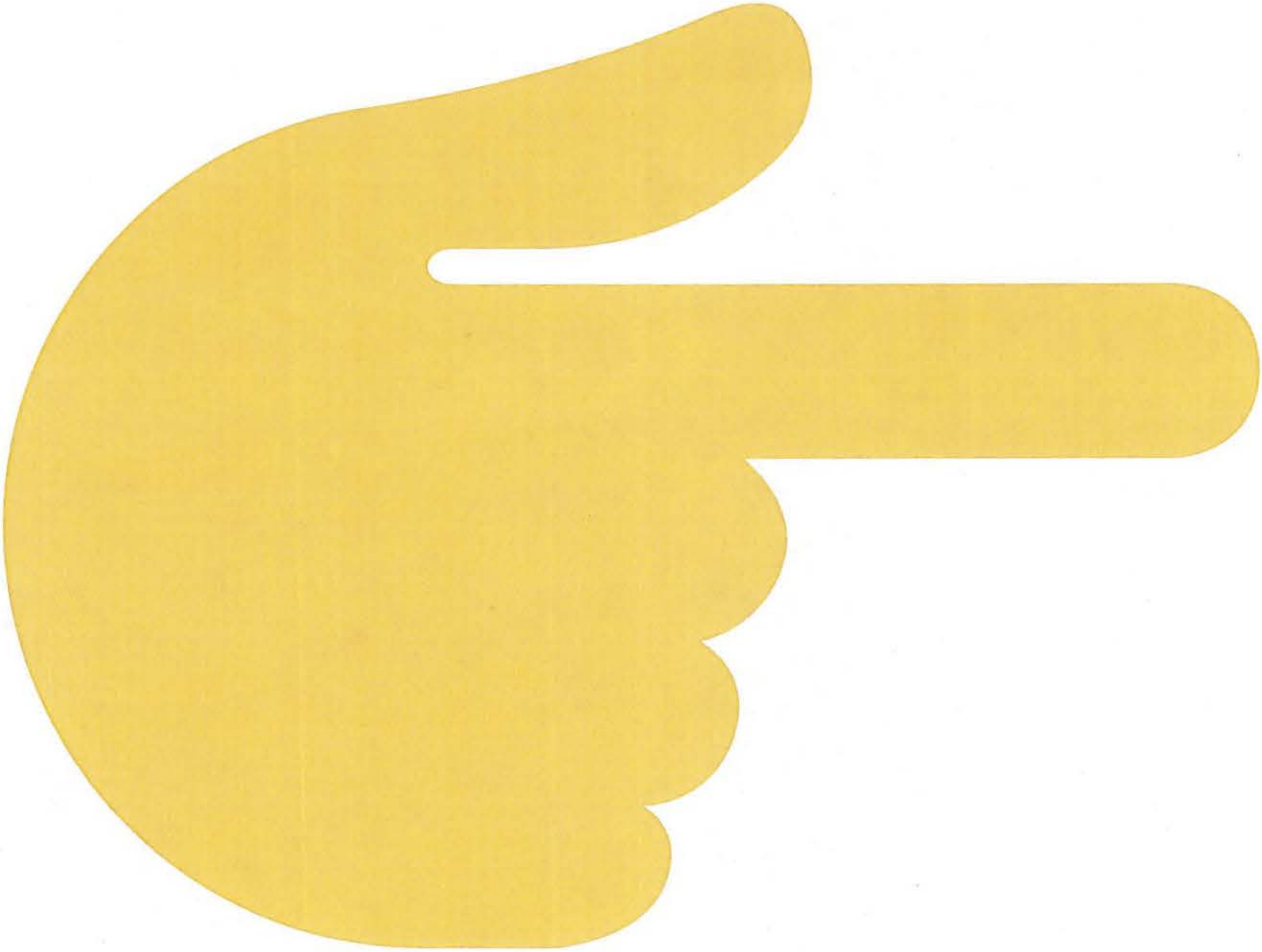


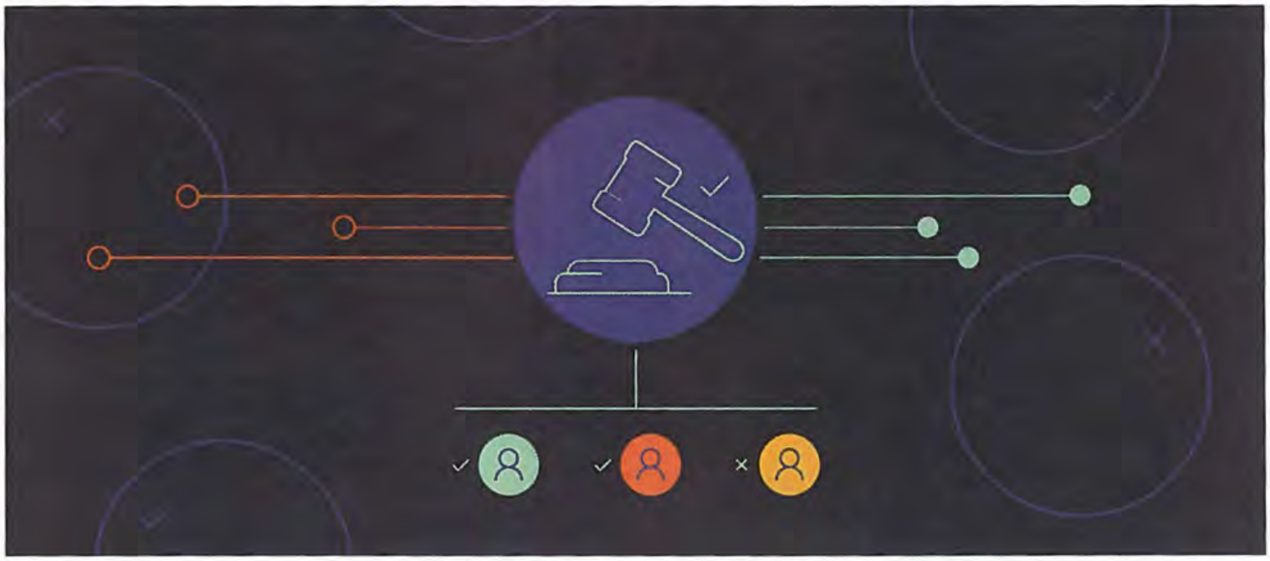
EXHIBIT 59

Dissecting the DAO: Web3 Ownership is Surprisingly Concentrated

blog.chainalysis.com/reports/web3-daos-2022/

Chainalysis Team

June 27, 2022



This blog is a preview of our State of Web3 Report. Sign up here to download your copy!

Decentralized autonomous organizations (DAOs) are a staple of web3. Internet-native and blockchain-based, DAOs are intended to provide a new, democratized management structure for businesses, projects, and communities, in which any member can vote on organizational decisions just by buying into the project.

At a high level, this is how DAOs work:

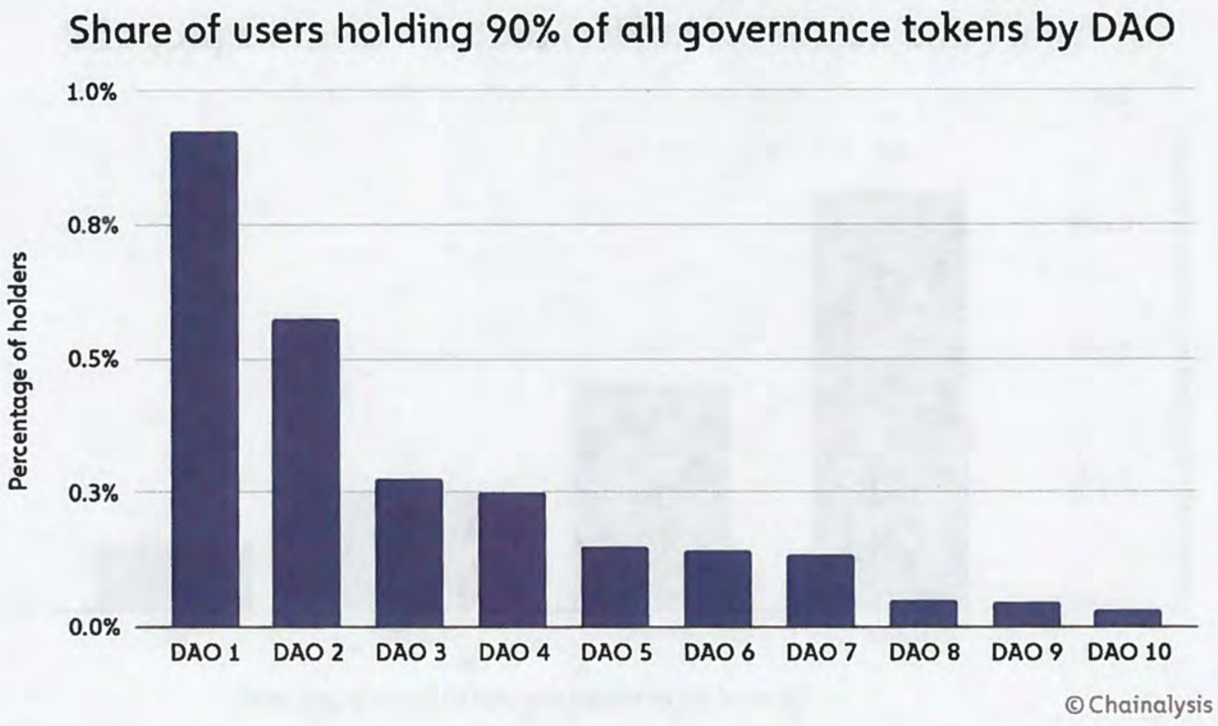
1. DAO founders create a new cryptocurrency, known as a **governance token**;
2. They distribute these tokens to users, backers, and other stakeholders;
3. Each token corresponds to a set amount of voting power within the organization. It also corresponds to a price on the secondary market, where it can be bought and sold at will.

While this process is often described as a way to decentralize power, governance token data suggests that DAO ownership is highly concentrated.

The concentration of governance token holdings

By analyzing the distribution of ten major DAOs' governance tokens, we find that, across several major DAOs, less than 1% of all holders have 90% of the voting power.

[redacted]



© Chainalysis

This has meaningful implications for DAO governance. For example, if just a small portion of the top 1% of holders worked together, they could theoretically outvote the remaining 99% on any decision. This has obvious practical implications and, in terms of investor sentiment, likely affects whether small holders feel that they can meaningfully contribute to the proposal process.

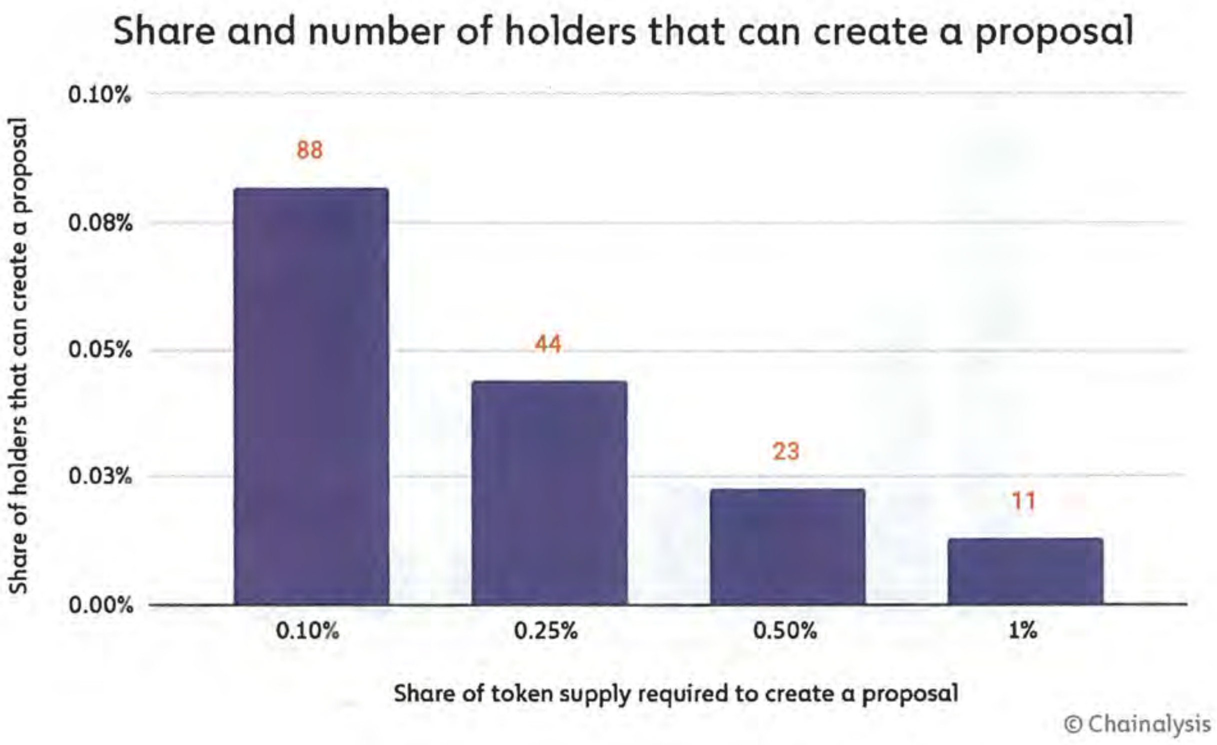
The impact of high concentration on DAO governance

For a governance token holder, there are three key governance actions. Voting is simple – any holder can do it. But what about creating a proposal? And what about passing it?

Per these ten DAOs proposal requirements, we find that:

1. A user must hold between 0.1% and 1% of the outstanding token supply to create a proposal.
2. A user must hold between 1% and 4% to pass it.

Using these ranges as lower and upper bounds, we find that **between 1 in 1,000 and 1 in 10,000 of these ten DAOs' holders have enough tokens to create a proposal**.



© Chainalysis

There are several tradeoffs at play here. If too many holders can create a proposal, the average proposal's quality may fall, and the DAO may be riddled with governance spam. But if too few can, the community may come to feel that "decentralized governance" rings false.

When it comes to single-handedly *passing* a proposal, **between 1 in 10,000 and 1 in 30,000 holders have enough tokens to do so**.

Overly concentrated voting power in DAOs can result in decision-making that seemingly contradicts the tenets of decentralization on which web3 is built. For instance, in June 2022, the DAO governing the Solana-based lending protocol Solend faced a problem: Solana's price was dropping, and if it fell much further, the protocol's biggest whale user would face a margin call that could render Solend insolvent and send roughly $20 million worth of Solana onto the market, potentially tanking the asset's price and upending the entire Solana ecosystem. The DAO called a vote to take control of the whale's account and liquidate its position through OTC desks, rather than the open market.

> Governance proposal SLND1 has passed.
>
> Special margin requirements for accounts that represent over 20% of borrows are now in effect.
>
> There will be a grace period for 3oSE…uRbE to reduce their leverage by themselves. pic.twitter.com/dsZhFRC8ZX
>
> — Solend (we're hiring!) (@solendprotocol) June 19, 2022

The proposal passed easily, with over 1.1 million "yes" votes to 30,000 "no" votes. However, more than 1 million of those votes came from a single user with enormous governance token holdings. Without their vote, the motion wouldn't have passed the 1% participation rate necessary for quorum.

The decision triggered a backlash from the cryptocurrency community, with many questioning how a platform could claim to be decentralized and then take control of a user's funds against their will. Following this, the Solend DAO voted again to invalidate the proposal, and the whale user eventually began to unwind their position. While the crisis was averted in this case, it raises questions about the ability of a DAO to act in the best interest of all participants when some voters control such an outsized share of governance tokens.

How do DAOs govern, exactly?

Actual governance processes vary enough from DAO to DAO that this question is best answered with examples. Let's start with the biggest one: Uniswap.

Example: Uniswap Governance

Uniswap is a decentralized exchange (DEX), and, like many DeFi protocols, it is governed by a DAO.

Anyone who holds Uniswap's governance token, UNI, is a member of this DAO. They can participate in governance by delegating their voting rights to their own or another's address, by publicizing their opinions, or by submitting their own proposal. The contents of these proposals vary widely: holders have recently voted on whether to finance a grant program, whether to integrate a new blockchain, and whether to reduce the governance proposal submission threshold.

But before someone can submit a proper proposal, their idea must pass the first two phases: temperature checks and consensus checks.

1. **The temperature check** determines whether there is sufficient community will to change the status quo. At the end of the two days, a majority vote with a 25,000 UNI yes-vote threshold wins.
2. **The consensus check** establishes formal discussion around a potential proposal. At the end of five days, a majority vote with a 50,000 UNI yes-vote threshold wins.

If both checks pass, an official governance proposal can be put to a vote. Then, there's a seven-day deliberation period to discuss the merits of this proposal on governance forums. If at the end of this period there are at least 40 million yes-votes with no-votes as a minority, the proposal has passed, and will be enacted after a two-day timelock.

Example: Dream DAO Governance

Not all DAOs function like Uniswap, but most at least run on similar infrastructure, using voting systems like Snapshot and chat servers like Discord. Dream DAO is no exception, though its mission and therefore its governance process is necessarily unique.

Dream DAO is an impact-oriented DAO created by 501(c)(3) charity Civics Unplugged and designed to provide diverse Gen Zers globally with the training, funding, and community they need to use web3 to improve humanity. Their governance process is run by holders of SkywalkerZ – NFTs that function as both governance tokens and fundraising incentives for anyone interested in donating to the program. For every SkywalkerZ NFT purchased by a donor, a new SkywalkerZ is reserved for a future Gen Zer to join as a voting member, thereby receiving power in the DAO without needing to pay. The purchaser of the NFT can apply to join the DAO and become a voting member as well, or they can leave it to the Gen Z student they've sponsored — either way, the NFT is theirs to keep.

By removing financial barriers from the process of participating in DAO governance, Dream DAO empowers its target audience – future Gen Z leaders – to influence decision-making, immerse themselves in web3, and leverage blockchain technologies positively.

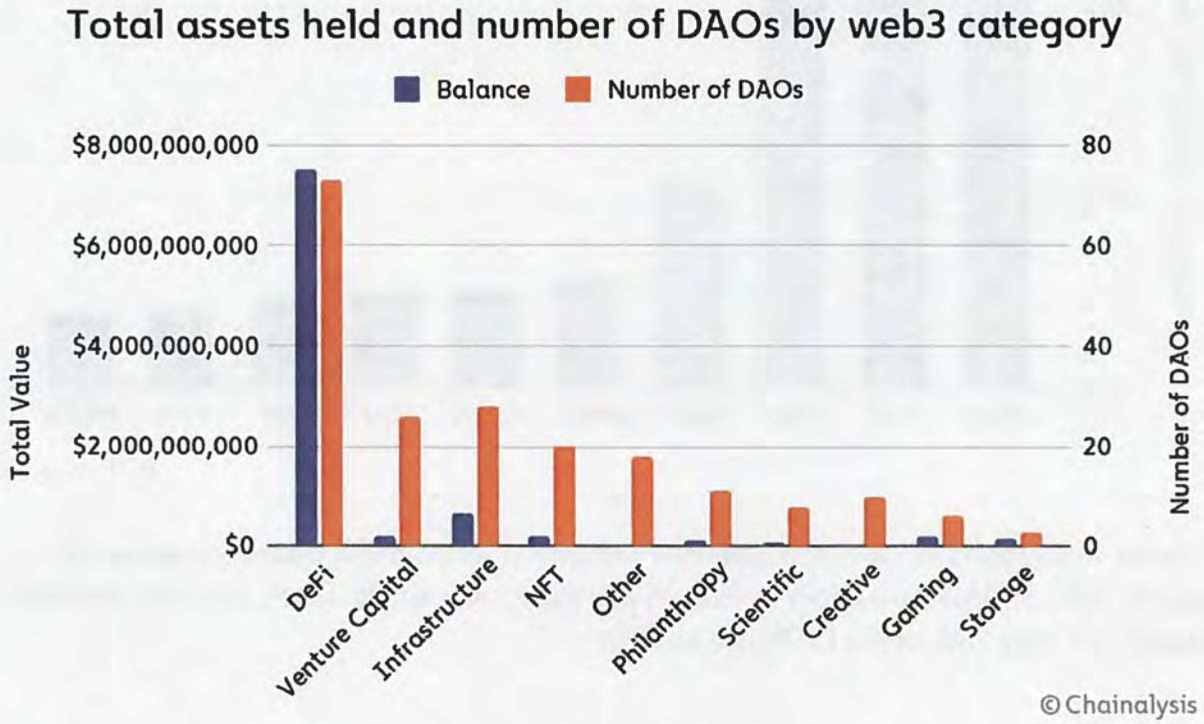
Where are DAOs most common and well-funded?

DAOs span the entire length of web3. They govern:

- **DeFi protocols** like Uniswap ($UNI) and Sushi ($SUSHI).
- **Social clubs** like Friends With Benefits ($FWB) and Bored Ape Yacht Club ($APE).
- **Grant-makers** like Gitcoin ($GTC) and Seed Club ($CLUB).
- **Play-to-earn gaming guilds** like Good Games Guild ($GGG) and Yield Guild Games ($YGG).
- **NFT generators** like Nouns (1 NFT = 1 vote).
- **Venture funds** like MetaCartel and Orange DAO.
- **Charities** like Big Green DAO and DreamDAO (1 SkywalkerZ = 1 vote).

- **Virtual worlds** like Decentraland ($MANA) and Sandbox ($SAND) .
- **And more**.

But in terms of raw numbers and treasury sizes, DeFi-related DAOs have a giant lead. The DeFi category accounts for 83% of all DAO treasury value held and 33% of all of the DAOs by count.

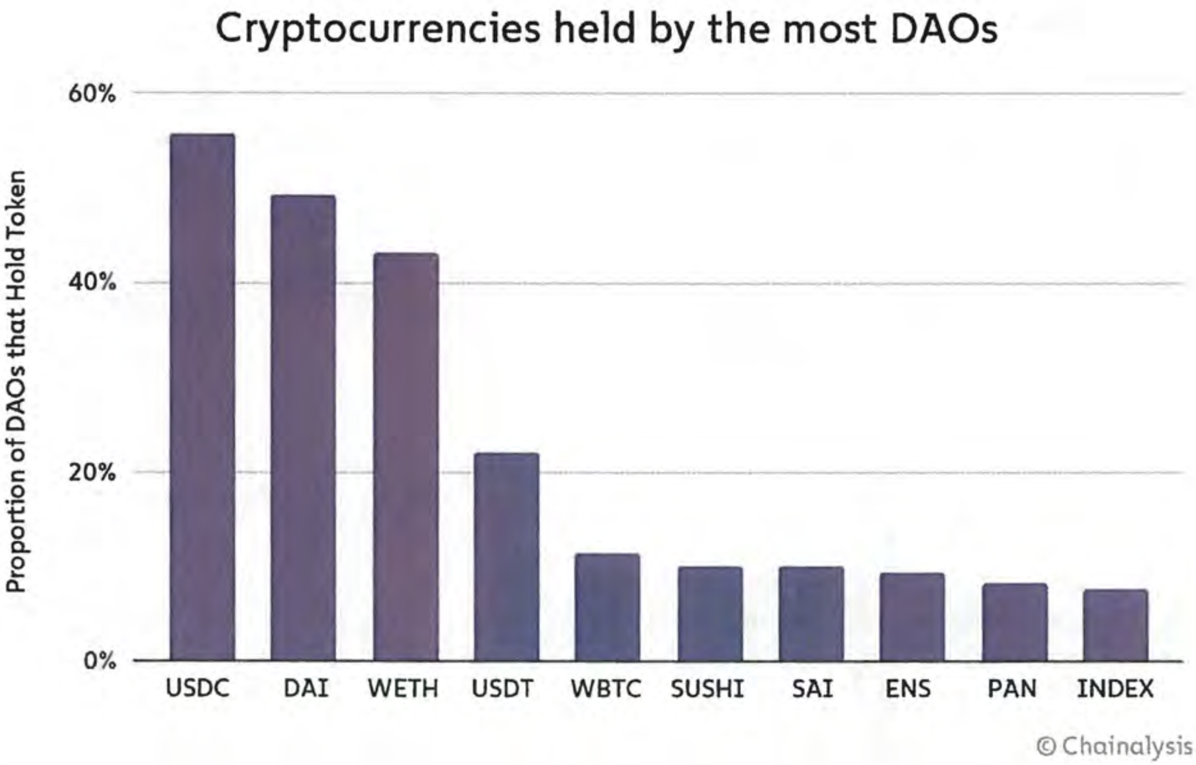


There are also a large number of DAOs focused on venture capital, infrastructure, and NFTs, suggesting that DAOs are appealing to investors, developers, and artists. Their on-chain treasuries, however, are relatively tiny.

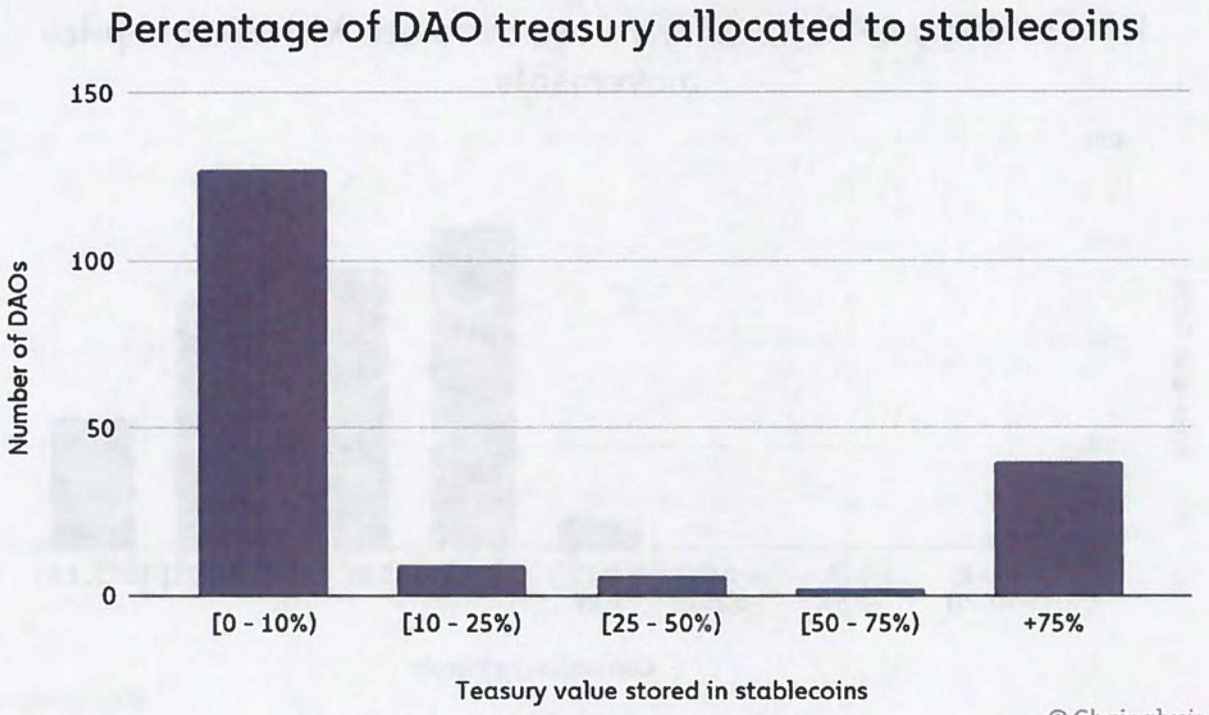
To be fair, the lines between these categories are blurry. Gaming DAOs often engage with NFTs, venture DAOs often provide funding to DeFi, and infrastructure DAOs support all of the above categories.

Treasury management: What assets do DAOs hold?

Even though DAOs vary in type and size, most of their on-chain treasuries hold similar cryptocurrencies. The most commonly held cryptocurrency is the stablecoin USD Coin (USDC), with over half of the 197 DAOs we analyzed holding a balance of USDC.



However, stablecoins seldom account for a majority of an on-chain treasury's value. On average, 85% of DAOs' on-chain treasuries are stored in a single asset, and that asset is a stablecoin in only 23% of the DAOs we studied.

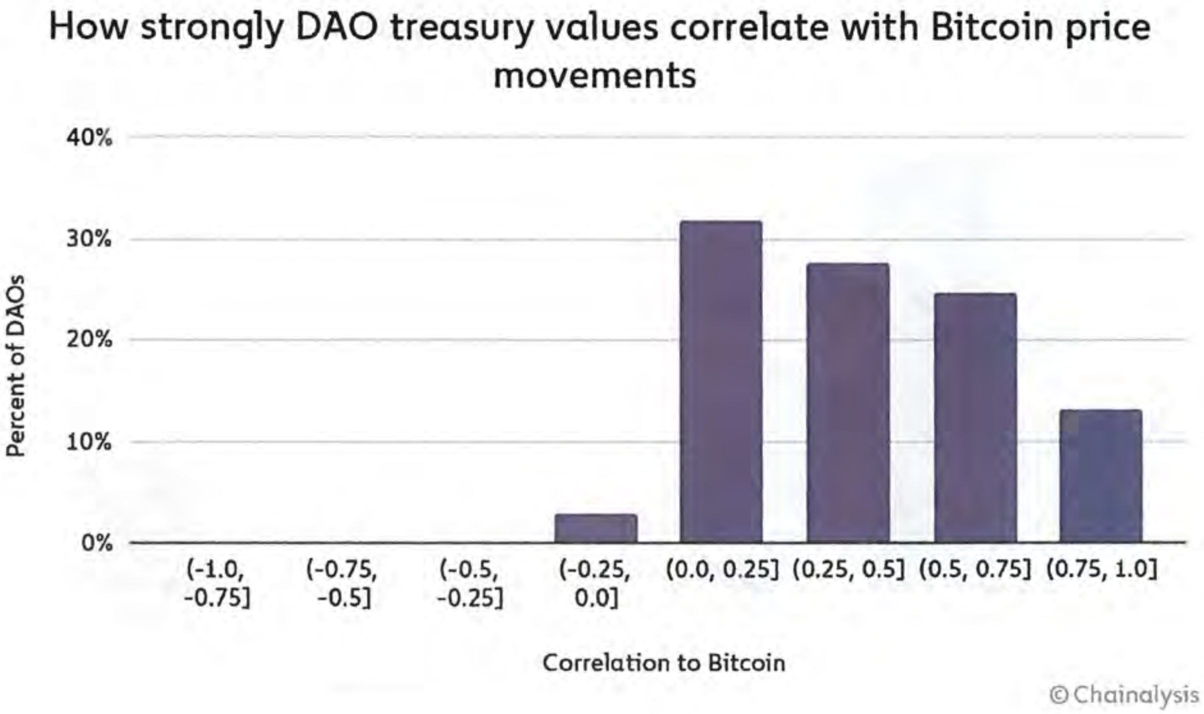


© Chainalysis

These on-chain treasuries are roughly as volatile as Bitcoin. By assuming DAOs' current holdings are their historical portfolios over the past year, we find that:

- The average DAO with assets over $1 million has an annualized volatility of 82%, versus 69% for Bitcoin.
- The average DAO with assets over $1 million suffered a maximum drawdown of 51% over the past year, compared to Bitcoin's drawdown of 72%.

DAO treasury values are also fairly correlated with Bitcoin price movements. 38% of on-chain DAO treasuries have correlations with Bitcoin that are between 0.5 and 1.00.

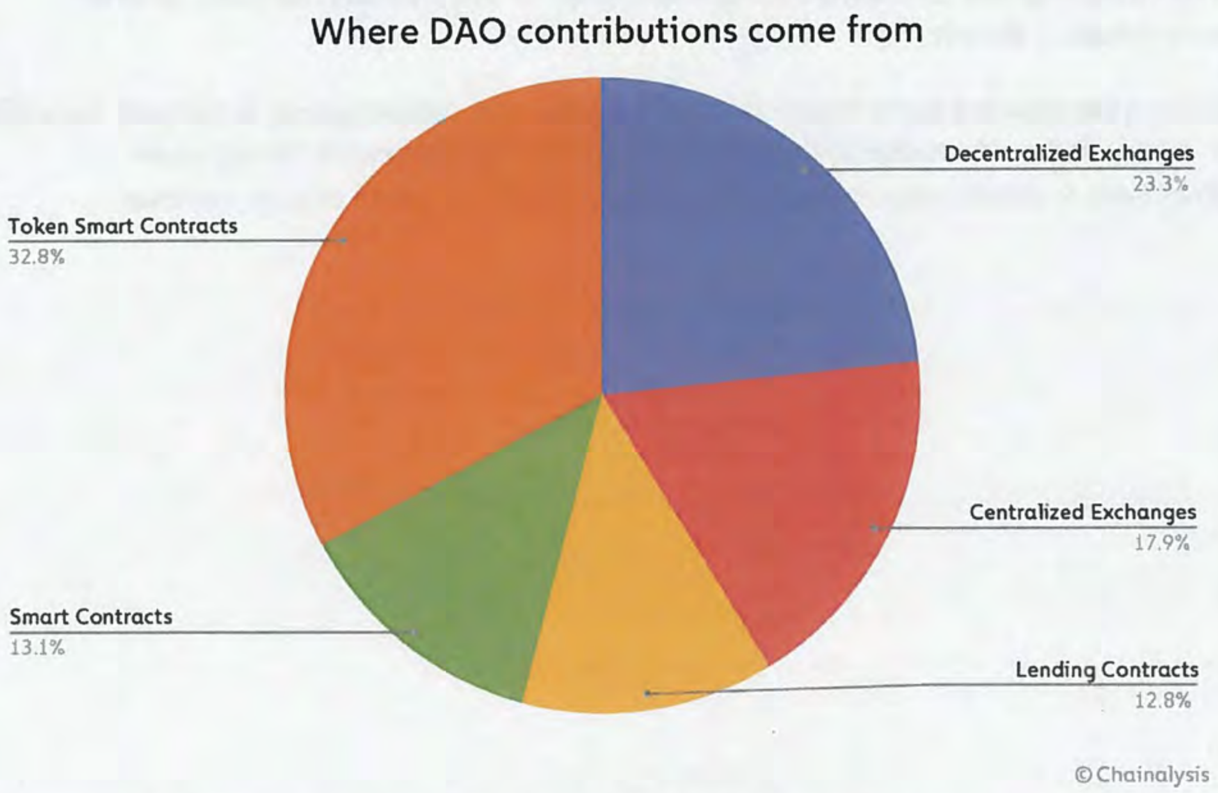


One of the most interesting areas of DAO treasury management that has yet to take off is in mergers and acquisitions (M&A). M&A makes sense for DAOs because it allows them to get into adjacent areas without having to develop internal tooling. As the DAO model matures, we suspect M&A will become more commonplace.

DAOs thus far have also been fairly limited in terms of the types of instruments they use and hold. For example, few DAOs to date have used loans or credit, perhaps due to their uncertain legal status. As DAOs mature, we are likely to see more standardized regulations, management strategies, and reporting practices.

Who contributes to DAOs?

While we don't collect demographic data about DAO participants, we can learn some things about DAO contributors using blockchain data.



Token smart contract = a project-specific ERC-20 or Layer 1 token contract

As one might expect, DAO participants are advanced users of cryptocurrency services. Only 17.9% of DAO treasury funds came from centralized services, while the remaining 82.1% originated at decentralized services. This suggests that most DAO contributors also engage with DeFi platforms and likely self-host their cryptocurrency.

The future of DAOs

As DAOs gain momentum, a cottage industry of tooling services and advocacy groups have emerged to help them grow and govern. Superdao streamlines DAO creation; Snapshot simplifies governance; and Coin Center advocates for the industry on Capitol Hill. As they continue to expand, it will be interesting to see what they can accomplish, what they will become, and to what extent they will achieve their goal to decentralize the ownership of the internet. With the proliferation of DAOs today, we'll have plenty of chances to see.

Download the State of Web3 Report

This website contains links to third-party sites that are not under the control of Chainalysis, Inc. or its affiliates (collectively "Chainalysis"). Access to such information does not imply association with, endorsement of, approval of, or recommendation by Chainalysis of the site *or its operators, and Chainalysis is not responsible for the products, services, or other content hosted therein.*

Chainalysis does not guarantee or warrant the accuracy, completeness, timeliness, suitability or validity of the information in this report and will not be responsible for any claim attributable to errors, omissions, or other inaccuracies of any part of such material.

EXHIBIT 62

Coin Center

Education

Cryptocurrency 101

Advanced Topics

What is multi-sig, and what can it do?

What is Bitcoin mining, and why is it necessary?

What is "open source" and why is it important?

What does "permissionless" mean?

What is "staking"?

What are mixers and "privacy coins"?

How does Tornado Cash work?

Policy and Regulation

Crypto Regulation FAQ

Key Concepts

51% Attack

Bitcoin

Blockchain

Blockchain Analysis

CoinJoin

How does Tornado Cash work?

by Alex Wade & Michael Lewellen & Peter Van Valkenburgh
August 25, 2022

1. Introduction

In August 2022, the US Treasury's Office of Foreign Assets Control (OFAC) sanctioned Tornado Cash, adding 45 Ethereum addresses to the Specially Designated Nationals (SDN) List of sanctioned persons.

This document aims to help the reader understand what Tornado Cash is, how it works, and what, exactly, was sanctioned. But before we jump into Tornado Cash, let's review a few key concepts around Ethereum, smart contracts, and decentralization.

2. Background: What is Ethereum, who are its users, what is a smart contract?

Ethereum is a cooperatively-run, global, transparent database. Through mutual effort, participants from all over the world maintain Ethereum's public record of addresses, which reference both user accounts and smart contract applications. These records work together much like the user accounts and software of a modern desktop computer, except that Ethereum is:

- Cooperatively-run: Ethereum's fundamental operation comes from the collective effort of its participants

Custody
Decentralized Exchange
Decentralized Markets
Energy Use
Ethereum
ERC-20 Tokens
Fraud
Hacking
Hard Fork
Key Storage Standards
Lightning Network
Payments
Micropayments
Mining
Mixers
Monero
Money Transmission
OFAC
Open Source
Permissionless
Privacy Coin
Smart Contracts
Store of Value
Unlicensed Money Transmission
Volatility
Wallets
Zcash

worldwide. No single party can make changes to how Ethereum works.

- Publicly-accessible: Anyone anywhere in the world can interact with Ethereum, its users, and its applications.
- Transparent: Anyone anywhere in the world can download and view all the information in Ethereum's database.

Anyone can be a user of Ethereum. Creating an account is simple, and does not require a phone number, email, or physical address. Instead, users install an application called a "wallet," which generates a unique identifier for that user called an "address" and a password-like number for authentication called a "private key." Much like a person with multiple email addresses, Ethereum's users can create and use as many addresses as they want. Unlike with email, however, Ethereum's users are not "customers" in the traditional sense. They are participants in a global computing system running on open-source software, which functions without third-party oversight. It is also important to note that Ethereum addresses controlled by the same user are not necessarily publicly linked to one another; they are simply unique identifiers that belong to the user who has the corresponding private key.

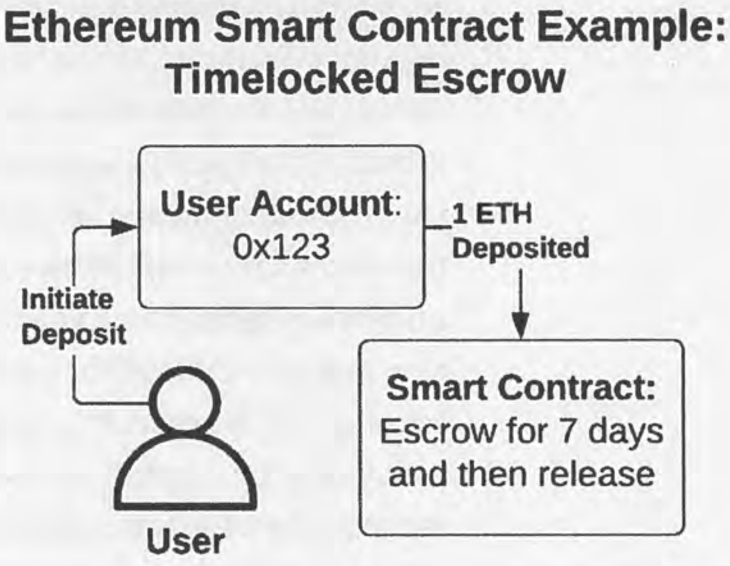
By sharing an address, users are able to receive tokens (*e.g.* crypto-assets like Ether) from anyone, anywhere in the world. Unlike a traditional payment service, sending and receiving tokens on Ethereum does not require an intermediary. Instead, the sender broadcasts their intent to transfer tokens, signs their message mathematically using the corresponding private key, and Ethereum's network collectively updates the global records of the sender and receiver addresses with the new balances. At no point in this process does a third party take custody of the tokens being transferred.

In addition to sending and receiving tokens, user accounts can interact with smart contracts, which are applications that extend the functionality of Ethereum. When developers program smart contracts, they decide what operations the smart contract will support and what rules those operations must follow. These rules and operations are written using code that is broadcast to Ethereum's network, just like the token transactions described above. Once a smart contract's code is added to Ethereum's records, it receives a unique address and can be interacted with by any user to automatically carry out the rules and operations it supports.

In essence, smart contracts are open-source applications that anyone can deploy to Ethereum. Just like the rest of Ethereum, smart contracts can be viewed and used by anyone, anywhere, and without relying on an intermediary.

Both people and smart contracts can have Ethereum addresses; the key difference is that when a person has an address they have the private key that controls any tokens sent to that address. That person will ultimately decide if and when any transactions are made with those tokens. When a smart contract has an address, the rules and operations written in the smart contract code control the tokens. They could be simple rules (*e.g.* automatically send the tokens back), or more complicated rules. There could be rules that include human operations and human decisions (*e.g.* send the tokens back if 3 out of 5 of these human-controlled addresses send a signed message saying they agree). The rules could also, however, be fully and permanently outside of any human being's control. In that case, so too are any tokens sent to that address until and unless the contract sends them back to some human according to the rules.

[redacted]


Ethereum Smart Contract Example: Timelocked Escrow
User Account: 0x123
1 ETH Deposited
Initiate Deposit
Smart Contract: Escrow for 7 days and then release
User

The User deposits 1 ETH token to be held in escrow for 7 days by the Smart Contract. After 7 days, the user can reclaim the tokens.

By default, smart contracts are immutable, which means they cannot be removed or updated by anyone once deployed. It is possible for the smart contract's developers to include (in the contract code) the ability to update functionality as a supported operation (*e.g.* this *human-controlled* address can rewrite the contract in the future). However, such an operation must be included in the smart contract's code prior to the smart contract's deployment (*i.e.* publication to the Ethereum network). Without the inclusion of updatability prior to deployment, a smart contract cannot be modified by anyone. It is also possible to revoke the ability to update functionality by transferring the permissions for this ability to a placeholder Ethereum address for which there is no corresponding private key. This placeholder is known as "the zero address." Once the ability to update a contract has been revoked, it cannot be reclaimed and the contract can no longer be changed.

Unlike traditional finance, Ethereum's records are completely transparent: anyone can download and view the balances and transaction history of its user accounts. Although user addresses are pseudonymous, if a real-world identity is linked to a user address, it becomes possible to trace that user's complete financial history. Ethereum's transparency is important for auditability (*e.g.* verifying that updates to records are valid). However, this transparency also makes it difficult for users to protect their personal information. By default, a record of a casual transaction today (*e.g.* paying for Wi-Fi at the airport) leads directly to records of earlier transactions, which may include any intimate, revealing, or sensitive transactions made by the same user long ago.

Among the many different applications smart contracts may support, they may also provide an avenue for users to regain the privacy they expect when interacting with financial systems. Central to that privacy is the use of smart contracts to break the public chain of records that would otherwise link your transaction today to every transaction you've ever made in the past. Enter Tornado Cash.

3. Tornado Cash: A smart contract application

Tornado Cash is an open source software project that provides privacy protection for Ethereum's users. Like many such projects, the name does not refer to a legal entity, but to several open source software libraries that have been developed over many years by a diverse group of contributors. These contributors have published and made Tornado Cash available for general use as a collection of smart contracts on the Ethereum blockchain.

As we will explain, some of these smart contracts have been sanctioned by OFAC. The core of Tornado Cash's privacy tools, however, make up a subset of the addresses sanctioned by OFAC: the Tornado Cash "pools." Each Tornado Cash pool is a smart contract deployed to Ethereum. Like other smart contracts, the pool contracts extend the functionality of Ethereum with specific operations that can be executed by any user of Ethereum according to the rules defined in the Tornado Cash contracts' code.

This section will describe how these pools work. In particular, it will describe the key innovation that enables these pools to function autonomously: an application of privacy-preserving mathematics known as "zero-knowledge cryptography."

Subsequent sections will describe the specific addresses sanctioned by OFAC, and what they do. An appendix at the end will list all of the sanctioned contracts and their salient features.

Tornado Cash Core Contracts: Pools

Tornado Cash pools are smart contracts that enable users to transact privately on Ethereum. When prompted by a user, pools will automatically carry out one of two supported operations: "deposit" or "withdraw." Together, these operations allow a user to deposit tokens from one address and later withdraw those same tokens to a different address. Crucially, even though these deposit and withdrawal events occur publicly on Ethereum's transparent ledger, any public link between the deposit and withdrawal addresses is severed. The user is able to withdraw and use their funds without fear of exposing their entire financial history to third parties.

In support of the deposit and withdrawal operations, these smart contracts encode strict rules that further define its functionality. These rules are automatically applied to the deposit and withdrawal operations to maintain a very important property shared by all Tornado Cash pools: **users can only withdraw the specific tokens they originally deposited.**

This property is enforced automatically for all the pool's operations, and ensures that Tornado Cash pools are entirely *non-custodial*. That is, a user who deposits and later withdraws tokens maintains total ownership and control over their tokens, even as they pass through the pool. At no point is the user required to relinquish control of their tokens to anyone.

A key principle of Tornado Cash pools is that a user's privacy is derived in large part from the simultaneous usage of the pool by many other users. If the pool had only a single user, it wouldn't matter that the link between the user's deposit and withdrawal addresses was severed: simple inference would make it obvious where the withdrawn tokens came from. Instead, pools are used by many users simultaneously. Think of it like a bank's safe deposit box room. Anyone can go and store valuables in a locked box in that room, and, assuming the locks are good, only the person with the key can ever get those valuables back. Security aside, however, this may or may not be privacy enhancing. If only one person is ever seen going into and out of the room, then we know any valuables in that room are theirs. If, on the other hand, many people frequently go into and out of the room, then we have no way of knowing who controls which valuables in which boxes. By guaranteeing the property that users can only withdraw tokens they originally deposited, many users can simultaneously use these pools with the assurance that no-one else will receive their tokens.

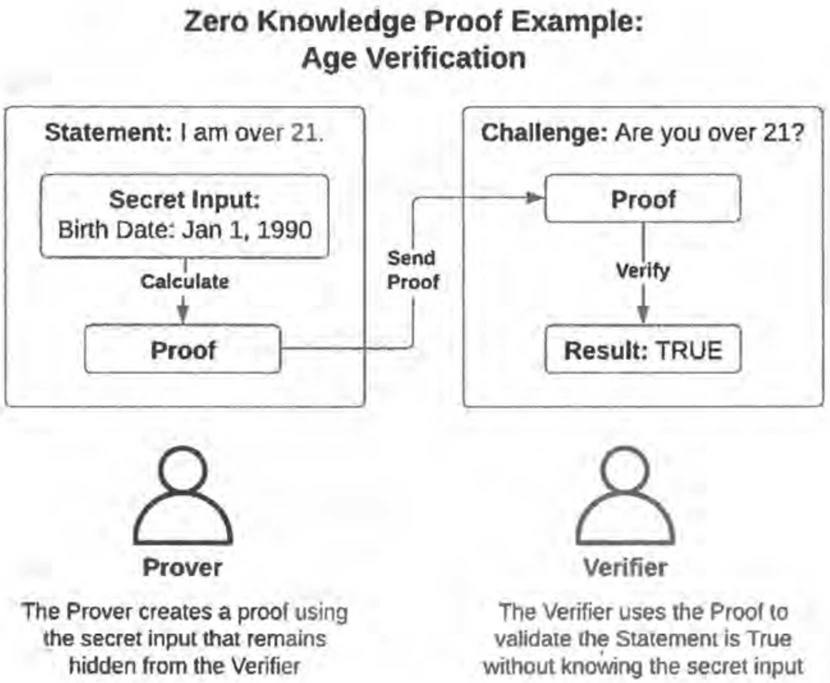
Traditionally, these assurances would be provided by a *custodial* service: a bank in the safe deposit box example, or a group of people running a "mixing service" in other common cryptocurrency arrangements. Mixing services like Blender.io directly accept tokens from their clients, aggregate and mix them, and then return the funds to their clients (often taking some fee in the process). During the intermediate aggregation and mixing stage, the funds in question are completely in the control of the operators of the mixing service and are commingled. At the final stage of the mixing process, a user would receive funds sourced directly from the myriad other users that also used the service.

In contrast, Tornado Cash pools have no custodial operator, and users only ever withdraw the tokens they originally deposited (rather than a mixture of tokens from the other users of the service). This is made possible because of important properties of the deposit and withdrawal operations, which are automatically carried out through the use of a privacy-preserving branch of mathematics called "zero-knowledge cryptography." This zero-knowledge cryptography is included in Tornado Cash's smart contract code, and forms the foundation on which the deposit and withdrawal operations function.

Zero-Knowledge Proofs

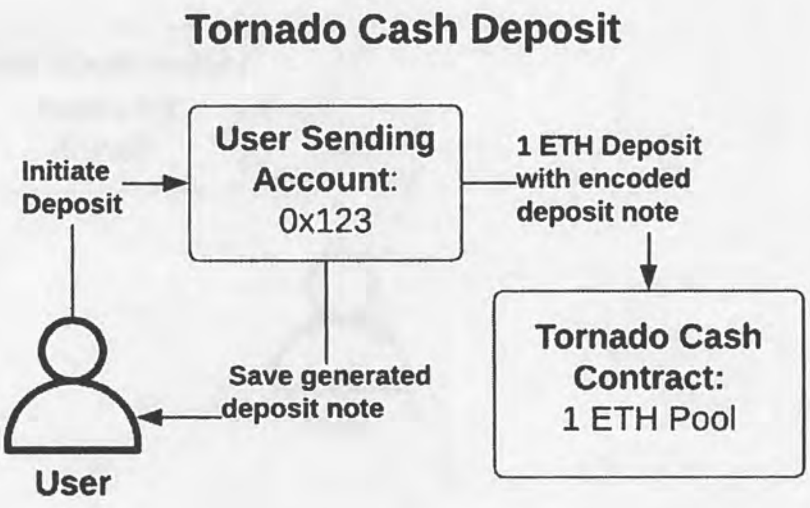
To recall an earlier point, Ethereum is transparent: anyone can view the transaction history and balance of any user account. Likewise, anyone can view the interaction history, balance, and code of a smart contract application. If a user prompts a smart contract to perform an operation, this interaction becomes a fact that is forever recorded in Ethereum's public records and can be recalled and inspected by anyone. So how is it that a user can deposit into a Tornado Cash pool and later withdraw to a different address without creating an obvious link to anyone observing Ethereum's public records?

The answer lies in *zero-knowledge proofs*. A zero-knowledge proof is a cryptographic method by which one party (the "prover") can prove to another party (the "verifier") that a given statement is true without the prover conveying any additional information apart from the fact that the statement is indeed true.


Zero Knowledge Proof Example:
Age Verification
Statement: I am over 21.
Secret Input:
Birth Date: Jan 1, 1990
Calculate
Proof
Send Proof
Challenge: Are you over 21?
Proof
Verify
Result: TRUE
Prover
The Prover creates a proof using the secret input that remains hidden from the Verifier
Verifier
The Verifier uses the Proof to validate the Statement is True without knowing the secret input

In the case of Tornado Cash, the "prover" is the user withdrawing tokens from the pool, while the "verifier" is one of the Tornado Cash pool contracts. When a user prompts the pool smart contract to withdraw their tokens, the user must supply the prompt with a zero-knowledge proof. The pool's code automatically checks the input proof, only processing a withdrawal if the proof is found to be valid. Exactly what statement is being proven by the user and how they create that proof is slightly more complicated, and requires a bit more detail on the deposit process.

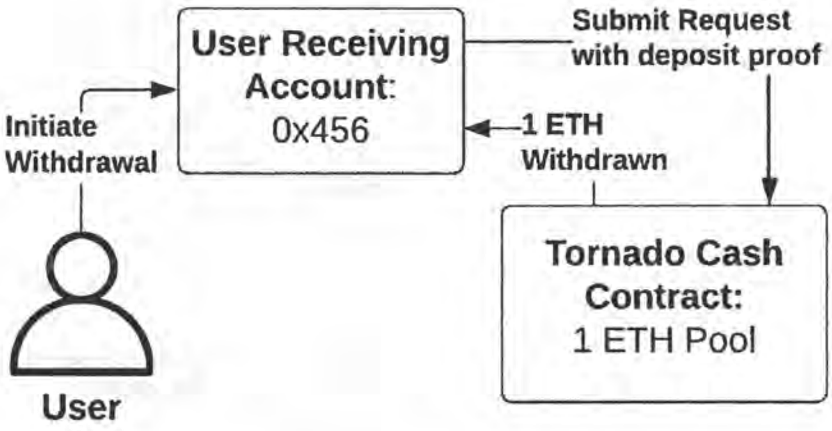
Pool Deposit Process


Tornado Cash Deposit
Initiate Deposit
User Sending Account: 0x123
1 ETH Deposit with encoded deposit note
Tornado Cash Contract: 1 ETH Pool
Save generated deposit note
User

When a user wants to deposit tokens, they first generate a “deposit note” (a long sequence of digits known only to the user). This is done privately on the user’s own computer, and is never shared publicly. Next, the user prompts the Tornado Cash pool contract to process the deposit. Along with this prompt, the user supplies a hash (or encoded form) of their deposit note and the tokens for deposit. The pool smart contract automatically records the encoded note as a new entry in a public list of other users’ encoded notes. At this point, the depositing user has completed the first part of the process, and retains the deposit note, which acts as a receipt to withdraw the tokens later.

Pool Withdrawal Process

Tornado Cash Withdrawal


User Receiving Account: 0x456
Submit Request with deposit proof
Initiate Withdrawal
1 ETH Withdrawn
Tornado Cash Contract: 1 ETH Pool
User

When a user is ready to withdraw their tokens, they first split their deposit note in half. One side acts like a "secret," and the other acts like a "lock." After that, the user prompts the Tornado Cash smart contract to withdraw. Along with the prompt, the user supplies:

- A *hash* (or encoded form) of the "lock"
- A *zero-knowledge proof*, generated using the "secret" and the "lock"

The pool smart contract uses these inputs to automatically verify – that is, *prove* – the following:

1. That the zero-knowledge proof was generated using the "secret." It is the exact same "secret" that corresponds to one of the existing encoded notes in the pool's public list of encoded notes (*i.e.* proving that the tokens being withdrawn were previously deposited by someone).
2. That the same proof also corresponds to the encoded form of the "lock" supplied with the proof (*i.e.* proving that the person who is withdrawing them must be the same person who deposited them).

3. That the submitted "lock" has not been submitted previously (*i.e.* the deposit in question has not already been withdrawn).

Assuming the proof is verified, the pool smart contract automatically:

1. Sends the user their tokens.
2. Records the encoded "lock" in a public list of other users' encoded locks, ensuring the same tokens cannot be withdrawn again.

Crucially, the above operations are carried out while the following is never revealed: which specific encoded note the proof corresponds to (*i.e.* who, among all of Tornado Cash's depositors, is now withdrawing).

Can Tornado Cash be removed or updated? If so, by whom?

As stated previously, for most readers, *Tornado Cash* is synonymous with a core subset of the Tornado Cash smart contracts: the Tornado Cash pools. The vast majority of these contracts are immutable. That is, they have no ability to be updated or removed by anyone. A complete list of sanctioned, immutable Tornado Cash pools can be found in Appendix A.

Note that many of these pools had, at one point, an "operator" role. The operator role was originally held by 0xDD4c...3384, aka *Gitcoin Grants: Tornado.cash*, another sanctioned address. This role afforded its holder two permissions:

- updateVerifier: Used to update the "verifier" used by the smart contract. In essence, this permission could be used to modify how the contract processed zero-knowledge proofs.

- changeOperator: Used to transfer the "operator" permission to another address, or revoke the "operator" permission entirely by transferring it to the zero address.

In May 2020, the updateVerifier permission was used in conjunction with the changeOperator permission as a final update to these Tornado Cash pools. This updated all pools' zero-knowledge proof processors to their final version, which incorporated the contributions of over 1,100 community participants. Additionally, this update revoked the "operator" permission by using changeOperator to transfer the permission to the zero address. In effect, the update performed in May 2020 cemented the community's preferences, and ensured no further changes could be made. Details on this process can be found here.

A handful of SDN-listed pools still have an "operator" permission. Of these, two belong to very old, now-unused versions of Tornado Cash. The remaining pools either have newer, immutable versions, or were used so little that they were likely overlooked during the May 2020 final update. Most of these remaining eight pools have never been used, and the ones that were used were only used once or twice within the past three years. A complete list of sanctioned, outdated Tornado Cash pools that retain the operator permission can be found in Appendix C.

Tornado Cash Auxiliary Contracts & Controls

Governance and TORN Token

The pool smart contracts represent the core of the Tornado Cash application, which remains immutable and uncontrolled by any party. However, OFAC's sanctions also include auxiliary smart contracts that provide coordination mechanisms for the continued maintenance and use of Tornado Cash by its community. Several of these contracts are unused today, belonging to older versions of Tornado Cash. A complete list of OFAC-sanctioned smart contracts that relate to Tornado Cash's community maintenance can be found in Appendix B.

The SDN List includes two primary contracts still in use today:

- *Tornado Cash (Router)*: References a registry of up-to-date Tornado Cash pools, consistent with the current version of Tornado Cash. Users may *optionally* choose to interact with Tornado Cash pools via the Router contract, which ensures their deposit and withdrawal operations are processed using up-to-date code.
- *Tornado Cash (Relayer Registry)*: References a registry of operators providing relay-assisted withdrawal services to users of Tornado Cash. Users may *optionally* elect to process their withdrawals via a relayer, which may afford additional privacy.

Unlike the pool smart contracts, the Router and Relayer Registry support some updatable functionality. However, the permission to update these contracts is held not by a human, but by another smart contract. This smart contract, also known as *Tornado Cash: Governance*, defines the rules and operations that determine how the Router and Relayer Registry may be updated.

In short, *Tornado Cash: Governance* provides that updates to these smart contracts are processed at the behest of the community, which holds public votes to determine what updates should occur, and when. Any holder of TORN tokens may participate in these votes. TORN is an ERC20-

token built on Ethereum that is expressly used by the community to vote on governance proposals. Any user of Ethereum may purchase TORN tokens and participate in this process.

Note that while this process allows the wider Ethereum community to participate in the development and maintenance of Tornado Cash, *no part of this process allows for the update or removal of Tornado Cash pool smart contracts.* Additionally, participating in the *Tornado Cash: Governance* process is *entirely optional*: users can use Tornado Cash pools without any involvement, oversight, or interaction with the *Tornado Cash: Governance* process.

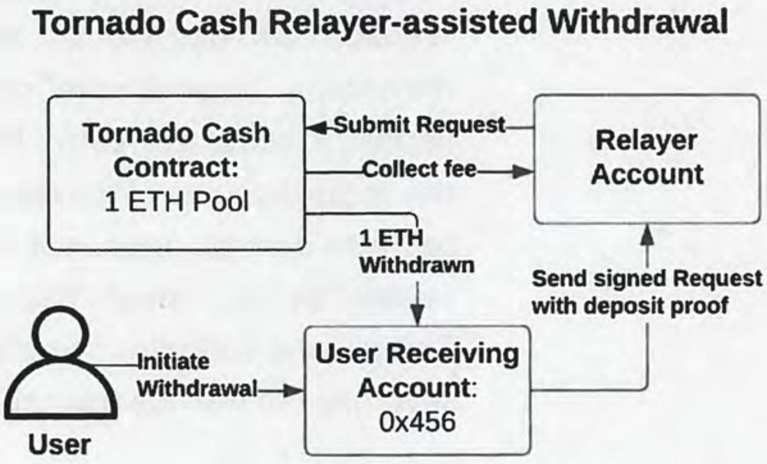
Although *Tornado Cash: Governance* and the TORN token contract are parts of the Tornado Cash software ecosystem, neither was added to OFAC's SDN List.

Relayers

As previously mentioned, "relayers" are independent operators that provide an *optional* service for Tornado Cash users.

By default, when users prompt the Tornado Cash pool contracts for withdrawal, the withdrawal account needs to already have Ether in order to pay the Ethereum network to process the smart contract's operations. However, sending Ether to the withdrawal account prior to withdrawal might create a link between the user's deposit and withdrawal accounts.

Relayers allow users to process withdrawals without needing to pre-fund their withdrawal accounts, which helps users maintain privacy when withdrawing.


Tornado Cash Relayer-assisted Withdrawal
Tornado Cash Contract: 1 ETH Pool
Submit Request
Collect fee
Relayer Account
1 ETH Withdrawn
Send signed Request with deposit proof
Initiate Withdrawal
User Receiving Account: 0x456
User

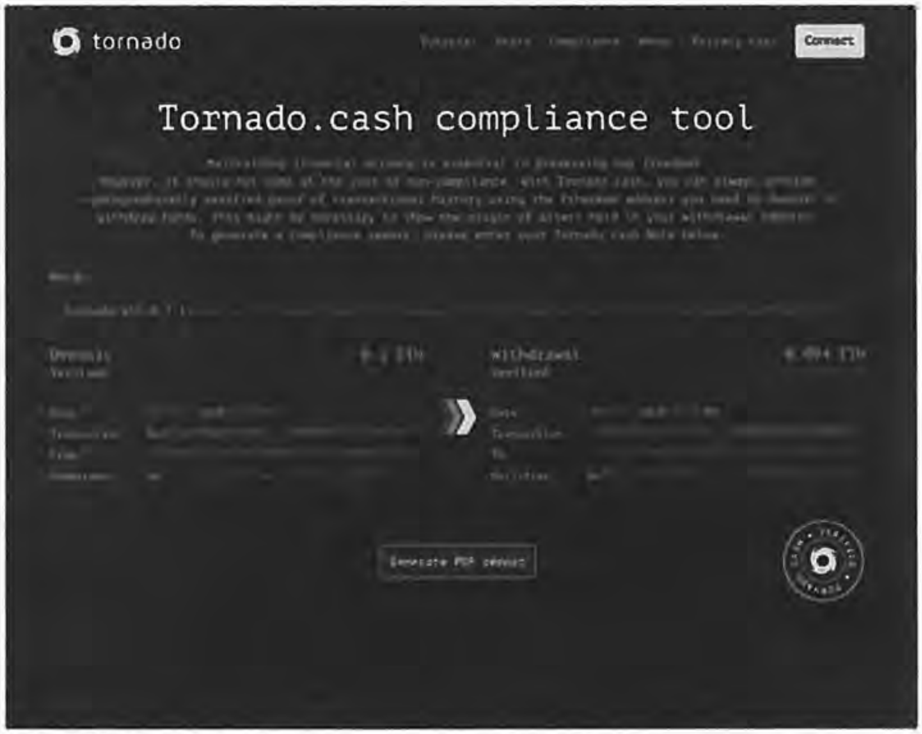
Users select a relayer from a public *Relayer Registry*, another sanctioned Tornado Cash smart contract. The user then uses their withdrawal account to sign a transaction authorizing the relayer-assisted withdrawal. The user sends this transaction to their selected relayer, who processes the withdrawal on their behalf, earning a fee in the process. Note that even though they process withdrawals on behalf of users, relayers never have custody over users' tokens; the smart contract ensures that withdrawn tokens are only ever sent to the user's withdrawal account.

OFAC has not specifically added any relayer addresses to the SDN List, but it has added the smart contract that contains a registry of relayers to the list.

Compliance Tool

Tornado Cash was built to enable Ethereum's users to reclaim their privacy. Rather than exposing their complete financial history, Tornado Cash gives users control over their personal information: both what is shared and with whom it is shared. However, maintaining privacy and preserving control over one's personal information does not need to come at the expense of non-compliance with legal obligations.

To this end, the developers of Tornado Cash created the *Tornado Cash Compliance Tool*. Users supply the tool with the original "deposit note" generated during the pool deposit process to create a PDF report that provides proof of the original source of the tokens. Although the public link between a user's deposit and withdrawal addresses was severed by the Tornado Cash pool contracts, the Compliance Tool allows users to selectively "undo" this severance to provide traceability to third parties.



The Compliance Tool is not a smart contract. However, just like the other software described in this article, the Compliance Tool is also not a service provided by Tornado Cash developers; it is an open-source tool that can be used by anyone.

Other Tornado Cash Smart Contracts and Addresses

Finally, two of the sanctioned addresses are donation addresses. These addresses were used in the past to raise money in support of the development of the privacy software that powers Tornado Cash. While some person or entity does control tokens sent to these addresses, those tokens are not, to our knowledge, being mixed or re-routed for privacy purposes. They are merely a gift from the sender in support of software development efforts performed by the recipient. A complete list of donation addresses sanctioned by OFAC can be found in Appendix D.

In general, while a minority of the contracts listed by OFAC do retain elements of human control, none of them are critical to the basic operation of Tornado Cash's privacy tools, and none of them take control of user tokens. The core privacy tools – the pool contracts – are outside of any individual or group's control; they are simply widely distributed computer code that is executed by the Ethereum network according to strict and unalterable rules.

Summary

In summary:

- The Tornado Cash smart contracts allow users to deposit and later withdraw their tokens to another address.
- Even though anyone can observe users deposit or withdraw tokens, they are not able to determine which withdrawals correspond to which deposits.
- These operations are defined as smart contract code and are carried out automatically without any intermediary or third party.
- Users retain control of their funds the whole time, and are only able to withdraw the tokens they originally deposit.
- No one controls the operation of these Tornado Cash smart contracts and no one has the ability to change their

operation in the future.

- Some OFAC-identified addresses retain a level of human control. However, these addresses are not core to the operation of the privacy tools found at the immutable addresses and they can not exercise control over any user tokens.

Appendix: Categorization of sanctioned addresses

These appendices list all addresses sanctioned by OFAC. They have been categorized according to the function they provide in the context of the Tornado Cash application.

Included with each address listed is the following information:

- *Name:* A name by which the address can be referenced. Note that these names do not come from the Tornado Cash developers, they come from Etherscan: a third party service whose website can be used to display information on the current state of Ethereum. The names listed are intended to be a handy reference, and do not necessarily reflect the views of the community or the Tornado Cash developers.
- *Description:* A short description of what each address refers to.

A: List of immutable Tornado Cash pools

- 0x12D66f87A04A9E220743712cE6d9bB1B5616B8Fc
  - *Name:* Tornado.Cash: 0.1 ETH

  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 0.1 ETH.
  - *Operator:* Revoked (operator set to zero address)
- 0x47CE0C6eD5B0Ce3d3A51fdb1C52DC66a7c3c2936
  - *Name:* Tornado.Cash: 1 ETH
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 1 ETH.
  - *Operator:* Revoked (operator set to zero address)
- 0x910Cbd523D972eb0a6f4cAe4618aD62622b39DbF
  - *Name:* Tornado.Cash: 10 ETH
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 10 ETH.
  - *Operator:* Revoked (operator set to zero address)
- 0xA160cdAB225685dA1d56aa342Ad8841c3b53f291
  - *Name:* Tornado.Cash: 100 ETH
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 100 ETH.
  - *Operator:* Revoked (operator set to zero address)
- 0xD4B88Df4D29F5CedD6857912842cff3b20C8Cfa3
  - *Name:* Tornado.Cash: 100 DAI
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 100 DAI.
  - *Operator:* Revoked (operator set to zero address)
- 0xFD8610d20aA15b7B2E3Be39B396a1bC3516c7144
  - *Name:* Tornado.Cash: 1000 DAI

[redacted]

  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 1000 DAI.
  - *Operator:* Revoked (operator set to zero address)
- 0x07687e702b410Fa43f4cB4Af7FA097918ffD2730
  - *Name:* Tornado.Cash: 10000 DAI 2
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 10000 DAI.
  - *Operator:* None (functionality not included)
- 0x23773E65ed146A459791799d01336DB287f25334
  - *Name:* Tornado.Cash: 100000 DAI
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 100000 DAI.
  - *Operator:* None (functionality not included)
- 0x22aaA7720ddd5388A3c0A3333430953C68f1849b
  - *Name:* Tornado.Cash: 5000 cDAI
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 5000 cDAI.
  - *Operator:* Revoked (operator set to zero address)
- 0xBA214C1c1928a32Bffe790263E38B4Af9bFCD659
  - *Name:* Tornado.Cash: 50000 cDAI
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 50000 cDAI.
  - *Operator:* Revoked (operator set to zero address)
- 0x03893a7c7463AE47D46bc7f091665f1893656003
  - *Name:* Tornado.Cash: 50000 cDAI 2

  - *Description:* A newer version of the 50000 cDAI Tornado Cash pool, which allows deposits and withdrawals in increments of 50000 cDAI.
  - *Operator:* None (functionality not included)
- 0x2717c5e28cf931547B621a5dddb772Ab6A35B701
  - *Name:* Tornado.Cash: 500000 cDAI 2
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 500000 cDAI.
  - *Operator:* None (functionality not included)
- 0xD21be7248e0197Ee08E0c20D4a96DEBdaC3D20Af
  - *Name:* Tornado.Cash: 5000000 cDAI
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 5000000 cDAI.
  - *Operator:* None (functionality not included)
- 0x4736dCf1b7A3d580672CcE6E7c65cd5cc9cFBa9D
  - *Name:* Tornado.Cash: 100 USDC
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 100 USDC.
  - *Operator:* Revoked (operator set to zero address)
- 0xd96f2B1c14Db8458374d9Aca76E26c3D18364307
  - *Name:* Tornado.Cash: 1000 USDC
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 1000 USDC.
  - *Operator:* Revoked (operator set to zero address)
- 0x169AD27A470D064DEDE56a2D3ff727986b15D52B
  - *Name:* Tornado.Cash: 100 USDT

[redacted]

  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 100 USDT.
  - *Operator:* Revoked (operator set to zero address)
- 0x0836222F2B2B24A3F36f98668Ed8F0B38D1a872f
  - *Name:* Tornado.Cash: 1000 USDT
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 1000 USDT.
  - *Operator:* Revoked (operator set to zero address)
- 0x178169B423a011fff22B9e3F3abeA13414dDD0F1
  - *Name:* Tornado.Cash: 0.1 WBTC
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 0.1 WBTC.
  - *Operator:* None (functionality not included)
- 0x610B717796ad172B316836AC95a2ffad065CeaB4
  - *Name:* Tornado.Cash: 1 WBTC
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 1 WBTC.
  - *Operator:* None (functionality not included)
- 0xbB93e510BbCD0B7beb5A853875f9eC60275CF498
  - *Name:* Tornado.Cash: 10 WBTC
  - *Description:* A Tornado Cash pool that allows deposits and withdrawals in increments of 10 WBTC.
  - *Operator:* None (functionality not included)

B: List of community-governed contracts

- 0xd90e2f925DA726b50C4Ed8D0Fb90Ad053324F31b
    - *Name:* Tornado.Cash: Router
    - *Description:* A contract that maintains a list of Tornado Cash pools, which can be used by users to route deposits and withdrawals to the correct Tornado Cash pool.
    - *Still in use*: Yes.
    - *Governance Controls:* No significant controls. The community may choose to withdraw any tokens sent to the Router, as the Router is not an intended recipient of tokens.
- 0x58E8dCC13BE9780fC42E8723D8EaD4CF46943dF2
    - *Name:* Tornado.Cash: Relayer Registry
    - *Description:* This contract allows anyone to register as a Tornado Cash relayer. Relayers provide an *optional* service for users to make gasless withdrawals.
    - *Still in use*: Yes.
    - *Governance Controls:* Updatable pending a community vote.
- 0x527653eA119F3E6a1F5BD18fbF4714081D7B31ce
    - *Name:* Tornado.Cash: Trees
    - *Description:* This contract holds a merkle tree (a kind of list) of all Tornado Cash deposit and withdrawal events.
    - *Still in use:* No, this is associated with an older version of Tornado Cash.
    - *Governance Controls:* Updatable pending a community vote.
- 0xCa0840578f57fE71599D29375e16783424023357

[redacted]

- *Name:* Tornado.Cash: L1 Helper
- *Description:* Allows users to designate deposited Ether to be bridged to a Tornado Cash pool located on the Gnosis Chain blockchain. This smart contract is not a major component of the Tornado Cash application.
- *Still in use:* Yes.
- *Governance Controls:* No significant controls. The community may choose to withdraw tokens sent to this contract, as it is not an intended recipient.

- 0x722122dF12D4e14e13Ac3b6895a86e84145b6967
  - *Name:* Tornado.Cash: Proxy
  - *Description:* An old version of *Tornado.Cash: Router*.
  - *Still in use:* No, this is associated with an older version of Tornado Cash.
  - *Governance Controls:* No significant controls following deprecation of this contract in Feb 2022. The community may choose to withdraw tokens sent to this contract, as it is not an intended recipient.
- 0x905b63Fff465B9fFBF41DeA908CEb12478ec7601
  - *Name:* Tornado.Cash: Old Proxy
  - *Description:* An old version of *Tornado.Cash: Router*.
  - *Still in use:* No, this is associated with an older version of Tornado Cash.
  - *Governance Controls:* No significant controls. The community may choose to withdraw tokens sent to this contract, as it is not an intended recipient.

C: List of outdated contracts that retain an operator permission

- 0x94A1B5CdB22c43faab4AbEb5c74999895464Ddaf
  - *Name:* Tornado.Cash: Mixer 1
  - *Description:* An old version of the Tornado Cash pools, unused today.
  - *Operator:* 0x8589427373D6D84E98730D7795D8f6f8731FDA16, aka *Tornado Cash: Donate*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to "enable" and "disable" the use of the contract. As of Oct 2019, the operator has "disabled" use of the contract.
- 0xb541fc07bC7619fD4062A54d96268525cBC6FfEF
  - *Name:* Tornado.Cash: Mixer 2
  - *Description:* An old version of the Tornado Cash pools, unused today.
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:* Updatable by the operator.
- 0xF60dD140cFf0706bAE9Cd734Ac3ae76AD9eBC32A
  - *Name:* Tornado.Cash: 10000 DAI
  - *Description:* Old/unused Tornado Cash pool that allows deposits and withdrawals in increments of 10000 DAI.
  - *Last used:* Feb, 2020

  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.
- 0xb1C8094B234DcE6e03f10a5b673c1d8C69739A00
  - *Name:* Tornado.Cash: 500000 cDAI
  - *Description:* Unused Tornado Cash pool that allows deposits and withdrawals in increments of 500000 cDAI.
  - *Last used:* Never used
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.
- 0xD691F27f38B395864Ea86CfC7253969B409c362d
  - *Name:* Tornado.Cash: 10000 USDC
  - *Description:* Unused Tornado Cash pool that allows deposits and withdrawals in increments of 10000 USDC.

  - *Last used:* Never used
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.
- 0xaEaaC358560e11f52454D997AAFF2c5731B6f8a6
  - *Name:* Tornado.Cash: 5000 cUSDC
  - *Description:* Old/unused Tornado Cash pool that allows deposits and withdrawals in increments of 5000 cUSDC.
  - *Last used:* May, 2020
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.
- 0x1356c899D8C9467C7f71C195612F8A395aBf2f0a
  - *Name:* Tornado.Cash: 50000 cUSDC

    - *Description:* Unused Tornado Cash pool that allows deposits and withdrawals in increments of 50000 cUSDC.
    - *Last used:* Never used
    - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
    - *Operator Controls:*
        - This contract cannot be removed by anyone.
        - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.

- 0xA60C772958a3eD56c1F15dD055bA37AC8e523a0D
    - *Name:* Tornado.Cash: 500000 cUSDC
    - *Description:* Unused Tornado Cash pool that allows deposits and withdrawals in increments of 500000 cUSDC.
    - *Last used:* Never used
    - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
    - *Operator Controls:*
        - This contract cannot be removed by anyone.
        - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.

- 0xF67721A2D8F736E75a49FdD7FAd2e31D8676542a
  - *Name:* Tornado.Cash: 10000 USDT
  - *Description:* Old/unused Tornado Cash pool that allows deposits and withdrawals in increments of 10000 USDT.
  - *Last used:* May, 2020
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.
- 0x9AD122c22B14202B4490eDAf288FDb3C7cb3ff5E
  - *Name:* Tornado.Cash: 100000 USDT
  - *Description:* Unused Tornado Cash pool that allows deposits and withdrawals in increments of 100000 USDT.
  - *Last used:* Never used
  - *Operator:* 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384, aka *Gitcoin Grants: Tornado.cash*
  - *Operator Controls:*
    - This contract cannot be removed by anyone.
    - The sole permission afforded to the Operator is the permission to update the "verifier" used by the

contract. In essence, the Operator may change how this contract processes zero-knowledge proofs.

D: List of donation addresses

- 0xDD4c48C0B24039969fC16D1cdF626eaB821d3384
  - *Name:* Gitcoin Grants: Tornado.cash
  - *Description:* A smart contract used to receive software development grants from the Gitcoin crowdfunding platform.
- 0x8589427373D6D84E98730D7795D8f6f8731FDA16
  - *Name:* Tornado.Cash: Donate
  - *Description:* A user address used to receive donations for software development. This address is not a smart contract.

Acknowledgements

- Shayan Eskandari (@sbetamc)
- Mikerah (@badcryptobitch)
- Banteg (@bantg)
- Wavey (@wavey0x)
- Hudson Jameson (@hudsonjameson)
- Kirill Pimenov (@kirushik)

- Mike Wawszczak (@mikedotwaves)
- Mary Mallor (@mmaller)
- M (@PopcornandWhiskey)
- Milo Murphy

Thank you to these independent researchers and advocates who donated their time and expertise to this effort.

Coin Center

Washington, DC

info@coincenter.org

@coincenter

Sitemap

About	Donate
Contact	Education
Blog	Reports
Testimony	Filings

Coin Center Policy Briefing

Receive periodic updates on policy research, testimony, and other Coin Center news.

Type your e | Subscribe

substack

EXHIBIT 63

Crypto Mixers and AML Compliance

blog.chainalysis.com/reports/crypto-mixers/

Chainalysis Team

August 23, 2022

What is a crypto mixer?

A crypto mixer is a service that blends the cryptocurrencies of many users together to obfuscate the origins and owners of the funds. Because Bitcoin, Ethereum, and most other public blockchains are transparent, this level of privacy is otherwise hard to achieve.

Why are crypto mixers used?

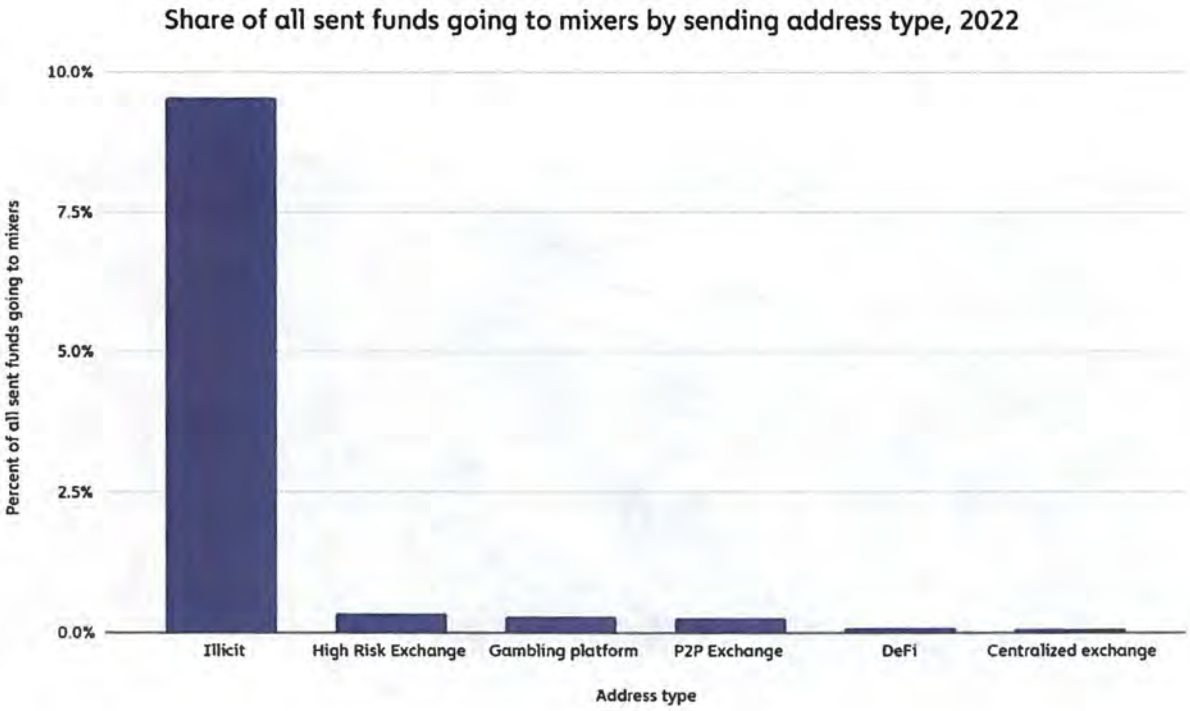
Financial privacy

Many use mixers out of a preference or need for privacy. Financial privacy is important, especially to those who live under oppressive regimes or who wish to make legal transactions anonymously.

Money laundering

A small percentage of crypto mixer users are cybercriminals. These criminals use mixers to obscure the connection between the crypto wallets they use to collect their illicit profits and the crypto wallets from which they transfer their funds to crypto-to-fiat exchanges. In this way, they aim to avoid triggering anti-money laundering alerts.

In July, we found that almost 10% of all cryptocurrencies held by illicit entities have been laundered through a mixer in 2022.



©Chainalysis

By comparison, only 0.3% of cryptocurrencies exposed to gray-area entities like gambling sites and high-risk exchanges have been mixed. This statistic falls to just 0.1% for cryptocurrencies exposed to regulated entities like centralized exchanges.

How crypto mixers work

Mixers collect, pool and pseudo-randomly shuffle the cryptocurrencies deposited by many users. Later, the funds are withdrawn to new addresses under the control of each user, minus a small service fee.

Most mixers make the deposited funds more difficult to track by letting users schedule their withdrawals in randomized amounts at randomized intervals. Others try to obfuscate the fact that a mixer is even being used; they typically do so by varying the transaction fee and the withdrawal address type.

The different types of crypto mixers

Most mixers fall under one of the following three categories, with the latter categories being the most novel and autonomous.

Centralized custodial mixers

Centralized custodial mixers, which emerged as early as 2011, temporarily take ownership of users' funds and are typically run by a single operator. Because this type of mixing service is both centralized and custodial, users face additional privacy risks. They are also often a target of law enforcement, as financial enforcement agencies treat them as unregistered money services businesses.

CoinJoins

A CoinJoin is a type of mixer commonly built into privacy wallets — meaning cryptocurrency wallets that pitch themselves on increased privacy — that combine users' coins with the coins of multiple other users in a single transaction. Users often repeat this process multiple times.

Unlike centralized mixers, CoinJoins are non-custodial, meaning they never actually hold users' funds.

Smart contract mixers

Like CoinJoins, smart contract mixers are non-custodial. But unlike CoinJoins, smart contract mixers don't combine users' funds in just one transaction. Instead, the user sends their funds to the mixer, receives a cryptographic note proving that they are the depositor, and then, whenever they'd like, sends the mixer that note to withdraw the funds to a new address. In the meantime, the cryptocurrencies are tumbled in a number of different ways.

Smart contract mixers often work with service providers called relayers, which can provide the ether necessary to pay the gas fees on mixer withdrawal transactions. This ensures that the user can withdraw their funds to new addresses with no transaction histories or connections to other services.

Are crypto mixers legal?

Despite their use by criminals, crypto mixers are not explicitly illegal in most jurisdictions. Whether they are compliant, however, is a different question.

In the United States, the Financial Crimes Enforcement Network (FinCEN) has confirmed that individuals and centralized businesses offering custodial mixing services must register as money transmitters under the Bank Secrecy Act (BSA), and have three key obligations:

1. register with FinCEN,
2. maintain an anti-money laundering and know-your-customer compliance program, and
3. meet all applicable reporting and record-keeping requirements.

We aren't aware of any custodial mixers currently following these rules. And given that privacy preservation is the main reason that many users interact with crypto mixers, it seems unlikely that one could implement these procedures and still retain their users.

Sanctions also matter for mixers. All mixers that want to do business in the U.S. must take measures to ensure they don't do business with sanctioned entities. And, as we'll cover below, even non-custodial smart contract-based mixers not covered by the BSA can be subject to sanctions designations, provided of course they aren't based in the U.S.

Enforcement actions against crypto mixers

Server seizures

In May 2019, the Dutch Fiscal Information and Investigation Service (FIOD), in close cooperation with Europol and authorities in Luxembourg, seized six servers controlled by the Bitcoin, Bitcoin Cash and Litecoin mixer Bestmixer.io.

Criminal charges

In April 2021, the Department of Justice (DOJ) arrested and charged the operator of Bitcoin Fog with money laundering, operating an unlicensed money transmitting business, and money transmission without a license.

In August 2021, the operator of the Bitcoin mixer Helix pleaded guilty to money laundering conspiracy and agreed to the forfeiture of more than 4,400 bitcoin, valued at more than $200 million at the time.

Sanctions designations

In May 2022, the U.S. Treasury's Office of Foreign Assets Control (OFAC) issued its first-ever sanctions on a crypto mixer, Blender.io, for its role in laundering funds stolen by North Korea in the hack of Ronin Bridge, a DeFi protocol linked to Axie Infinity.

In August 2022, OFAC sanctioned the most popular Ethereum mixer, Tornado Cash, for its role in laundering funds stolen by North Korean-linked hackers in the attacks on the Ronin and Harmony bridges.

Civil penalties

In October 2020, FinCEN penalized the operator of the Bitcoin mixers Helix and Coin Ninja $60 million dollar civil money penalty for operating two unregistered money services businesses (MSB).

How Chainalysis can help

Our blockchain forensics product, Chainalysis Reactor, has the most extensive mixer coverage and analytics tooling in the industry. Financial privacy is valuable, but so is consumer safety: our data shows that some 25% of mixed funds come from illicit addresses, and cybercriminals associated with hostile governments have mixed some of the largest sums. It's therefore important that stakeholders in the public and private sectors work together to address these risks – and use best-in-class data to inform their decisions.

To that end, Chainalysis's cryptocurrency compliance software, blockchain forensics tools, and government solutions teams are ready to help.

This material is for informational purposes only, and is not intended to provide legal, tax, financial, or investment advice. Recipients should consult their own advisors before making these types of decisions. Chainalysis has no responsibility or liability for any decision made or any other acts or omissions in connection with Recipient's use of this material.

EXHIBIT 72

🔊 Lees voor ▶

Arrest of suspected developer of Tornado Cash

Publicatiedatum 12-08-2022, 10:00

On Wednesday 10 August, the FIOD arrested a 29-year-old man in Amsterdam. He is suspected of involvement in concealing criminal financial flows and facilitating money laundering through the mixing of cryptocurrencies through the decentralised Ethereum mixing service Tornado Cash. Multiple arrests are not ruled out. These advanced technologies, such as decentralised organisations that may facilitate money laundering are receiving extra attention from the FIOD. Also in the cryptocurrency domain, the FIOD stands for a safe financial Netherlands and investigates with effect and impact. Today the suspect is brought before the examining judge.

Tornado Cash is a mixing service for cryptocurrencies. The online service makes it possible to conceal the origin or destination of cryptocurrencies. The (criminal) origin of the cryptocurrencies is often not or hardly checked by such mixing services. Users of a mixing service mostly do this to increase their anonymity.

In June 2022 the het Financial Advanced Cyber Team (FACT) of the FIOD started an criminal investigation against Tornado Cash, that is offered on the Internet by means of a decentralised autonomous organisation (DAO). A DAO is an organisation that does not have a hierarchical decision-making process, for example by a board. Instead, decisions are made on the basis of members' votes. The decisions are recorded in a programming code in a so-called 'smart contract'. The operation of the DAO also takes place using these smart contracts.

FACT suspects that through Tornado Cash has been used to conceal large-scale criminal money flows, including from (online) thefts of cryptocurrencies (so-called crypto hacks and scams). These included funds stolen through hacks by a group believed to be associated with North Korea. Tornado Cash started in 2019 and according to FACT it has since achieved a turnover of at least seven billion dollars. Investigations showed that at least one billion dollars' worth of cryptocurrencies of criminal origin passed through the mixer. It is suspected that persons behind this organisation have made large-scale profits from these transactions.

Since Monday 8 August 2022 Tornado Cash has been placed by the US government on the OFAC sanctions list of America.

The investigation is led by the Public Prosecutor's Office for serious fraud, environmental crime and asset confiscation.

Tackling money laundering as a priority

Tackling money laundering is a priority for the government, as it is of great importance in effectively combating all forms of serious crime. By disguising the criminal origin of the proceeds of crime, perpetrators are able to remain beyond the reach of the investigative authorities and enjoy their undisturbed criminal earnings.Money laundering via decentralised autonomous organisations is a new phenomenon that is receiving explicit attention from the FIOD.

For more information: Wietske Visser, 06 – 18 60 77 61

Categorie: Nieuws

EXHIBIT 86

tornado-repositories / **tornado-classic-ui** Public

forked from tornadocash-community/tornado-classic-ui

<> Code | Pull requests 1 | Actions | Projects | Security | Insights

master ▾ ···

tornado-classic-ui / **LICENSE**

tornado-repositories/tornado-classic-ui is licensed under the

MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	ⓘ License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		

This is not legal advice. Learn more about repository licenses.

dan1kov init — History

1 contributor

21 lines (17 sloc) | 1.04 KB ···

```
MIT License

Copyright (c) 2022 Tornado Cash

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
```

[redacted]

```

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

EXHIBIT 89

ALL ARTICLES › UNIVERSITY

Crypto Tokens vs Coins — What's the Difference?

Are crypto tokens and coins the same thing? Not exactly. Here we explain how to tell a coin from a token, and their different uses.

JUN 20, 2022 | BEGINNER

Key Takeaways

Coins

- A crypto coin is a form of digital currency that's often native to its blockchain; it stores value and acts as a medium of exchange

- Coins can be mined through proof of work (PoW) or earned through proof of stake (PoS)
- Examples include Bitcoin (BTC), Ether (ETH), and Cardano (ADA)

Tokens

- A crypto token is built for a decentralised project on an existing blockchain (usually Ethereum, the most popular blockchain for decentralised projects to build upon)
- A token represents an asset or offers holders certain platform-specific features
- Tokens offer functions, including utility, security, and governance
- Examples include Cronos (CRO), Very, Very Simple Finance (VVS), and Uniswap (UNI)

Token vs Coin: What is the Difference?

While many people use the phrases 'crypto coin', 'crypto token', and 'cryptocurrency' interchangeably, they're not the same thing. Though coins and tokens use distributed ledger technology (also known as blockchain technology), there are some significant differences between a coin and a token.

The TLDR is: Crypto coins are a form of digital currency that are often native to a blockchain, with the main purpose of storing value and working as a medium of exchange.

Crypto tokens are digital assets that are built on top of an existing blockchain (using smart contracts) and can serve a wide variety of functions, from representing a physical object to granting access to platform-specific services and features.

What is a Crypto Coin?

Crypto coins are native to their own blockchain. The Bitcoin blockchain coin is BTC. The Ethereum blockchain has ETH. And the Litecoin blockchain uses LTC. These crypto coins are primarily designed to store value and work as a medium of exchange, similar to traditional currencies. This is why crypto coins are also referred to as cryptocurrencies.

One of the other unique things about coins is the way they come into being. Generally, crypto coins are either mined using a proof of work (PoW) consensus mechanism or earned via a proof of stake (PoS) mechanism.

What are Coins Used for?

When Bitcoin was created, it was envisioned as a replacement for traditional fiat currencies.

Along with other crypto coins, it was **designed to work in the same ways as paper money** and metal coins, meaning it can be used for many of the things normally used with US dollars or Euros, including:

- Storing value
- Exchanging for other currencies
- Paying for goods and services
- Transferring to others

In addition to these traditional uses, some crypto coins can also take advantage of smart contract technology to offer additional features. For example, DASH is an altcoin that acts as a cryptocurrency but also gives holders the ability to vote in a decentralised autonomous organisation (DAO).

Popular Crypto Coins

- **Bitcoin (BTC)** was launched in early 2009 by the mysterious 'Satoshi Nakamoto', Bitcoin is the first and most well-known crypto coin in the world. Its head start has allowed it to become the most valuable cryptocurrency
- **Ether (ETH)** is one of the most popular crypto coins around and more than just a cryptocurrency. Thanks to the creation and implementation of smart contracts, Ethereum has become home to thousands of blockchain projects and NFTs. In some ways, it's the backbone of the blockchain revolution
- **Cardano (ADA)** is an open-source and decentralised blockchain platform that was one of the first to run on a PoS consensus, gaining a rep as a green crypto coin. Cardano was founded in 2015 by Ethereum co-founder Charles Hoskinson and facilitates peer-to-peer (P2P) transactions with its coin ADA

What are Tokens?

Like crypto coins, crypto tokens are designed using blockchain technology; however, crypto tokens aren't native to a blockchain. Instead, they're built on top of it, often utilising smart contracts to fulfil a variety of purposes.

While crypto coins mimic traditional currencies, crypto tokens are more like assets or even deeds. A crypto token can represent a share of ownership in a DAO, a digital product or NFT, or even a physical object. Crypto tokens can be bought, sold, and traded like coins, but they aren't used as a medium of exchange.

To use a real-world example, crypto tokens are more like coupons or vouchers, while crypto coins are like dollars and cents.

There are numerous types of crypto tokens:

Some governance tokens offer holders voting rights in a DAO.

Utility tokens may provide access to certain services or products developed by the token issuer.

Security tokens act like traditional securities and are even treated the same by many governmental agencies.

What are Tokens Used for?

Most crypto tokens are designed to be used within a blockchain project or dapp. Unlike crypto coins, tokens aren't mined; they are created and distributed by the project developer. Once tokens are in the hands of purchasers, they can be used in countless ways.

Popular Crypto Tokens

- **Filecoin** (FIL) and **Arweave** (AR) give users the ability to spend their utility tokens for the privilege of storing data on their decentralised network, pushing the concept of cloud storage to its full potential
- **Axie Infinity,** one of the best-known play-to-earn (P2E) on the market, features a utility token called **Smooth Love Potions** (SLP). By earning or purchasing SLP, players can perform exclusive in-game tasks

- **Cronos** (CRO) is the utility token for the Crypto.com ecosystem. CRO can be used to pay fees on the platform or staked for various benefits, and it allows token holders to trade crypto tokens for fiat at a reduced price

What About Stablecoins? Are They Coins or Tokens?

Stablecoins are cryptocurrencies tied to specific assets. They are a bit of a misnomer, as most of them are actually ERC-20 tokens (i.e., they operate on the Ethereum blockchain through a smart contract). So why are they called stable*coins*? The name lends itself to their primary function of being a medium of exchange.

Take **USD Coin (USDC),** for example. It is a smart-contract-based stablecoin (i.e., it doesn't have its own chain and is an ERC-20 token). It is backed by US dollars, held by the company that issues the token, to maintain the value of every USDC at US$1.

Conclusion

The question of whether to buy coins or tokens is largely dependent upon a holder's goals. Both can be purchased in the Crypto.com App or on the Crypto.com Exchange with low fees and best execution prices.

Browse our data and descriptions of thousands of coins and tokens on Crypto.com Price.

Due Diligence and Do Your Own Research

All examples listed in this article are for informational purposes only. You should not construe any such information or other material as legal, tax, investment, financial, or other advice. Nothing contained herein shall constitute a solicitation, recommendation, endorsement, or offer by Crypto.com to invest, buy, or sell any coins, tokens, or other crypto assets. Returns on the buying and selling of crypto assets may be subject to tax, including capital gains tax, in your jurisdiction.

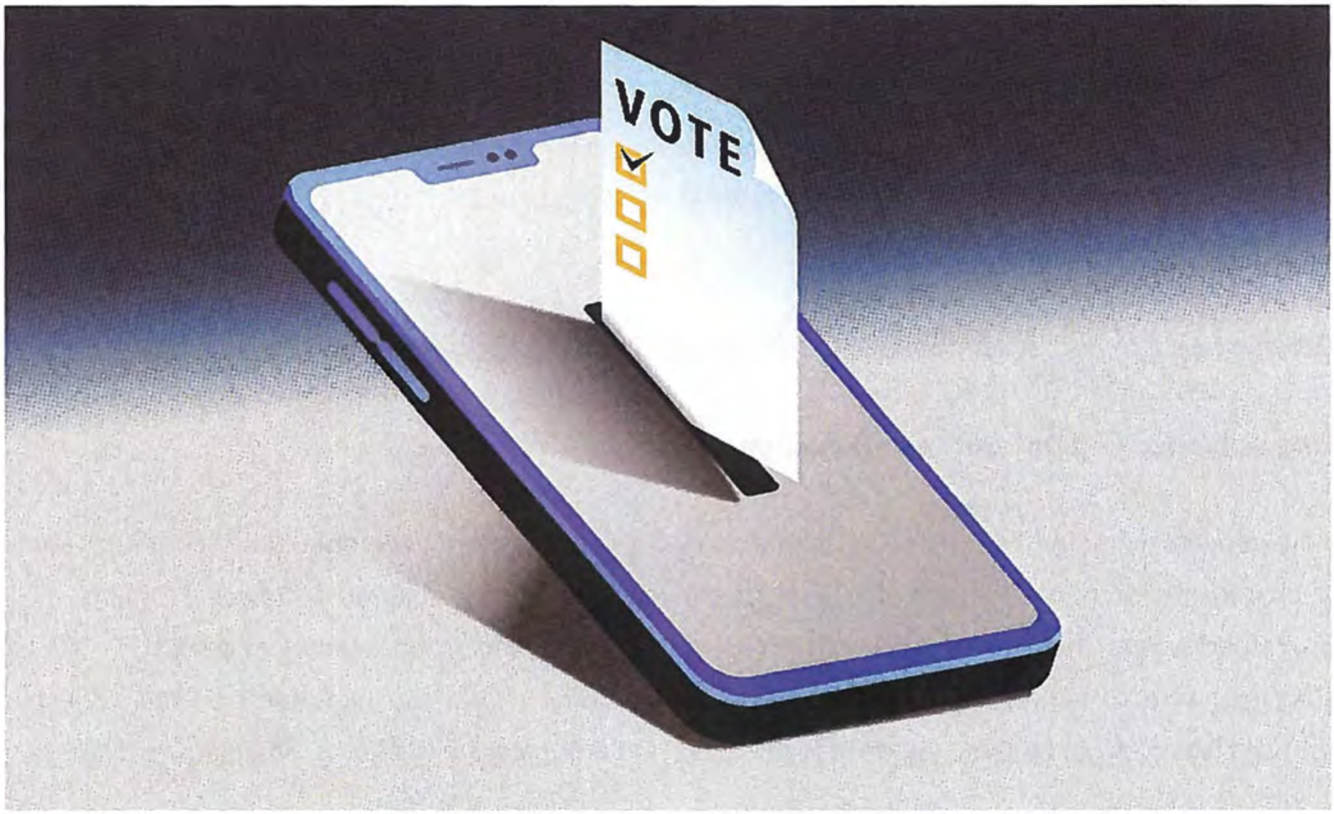
Past performance is not a guarantee or predictor of future performance. The value of crypto assets can increase or decrease, and you could lose all or a substantial amount of your purchase price. When assessing a crypto asset, it's essential for you to do your research and due diligence to make the best possible judgement, as any purchases shall be your sole responsibility.

Tags

ALTCOINS / COINS / COINS & TOKENS / CRYPTO101 / CRYPTOCURRENCIES / TOKENS

Share with Friends

RELATED CONTENT



UNIVERSITY / COINS & TOKENS

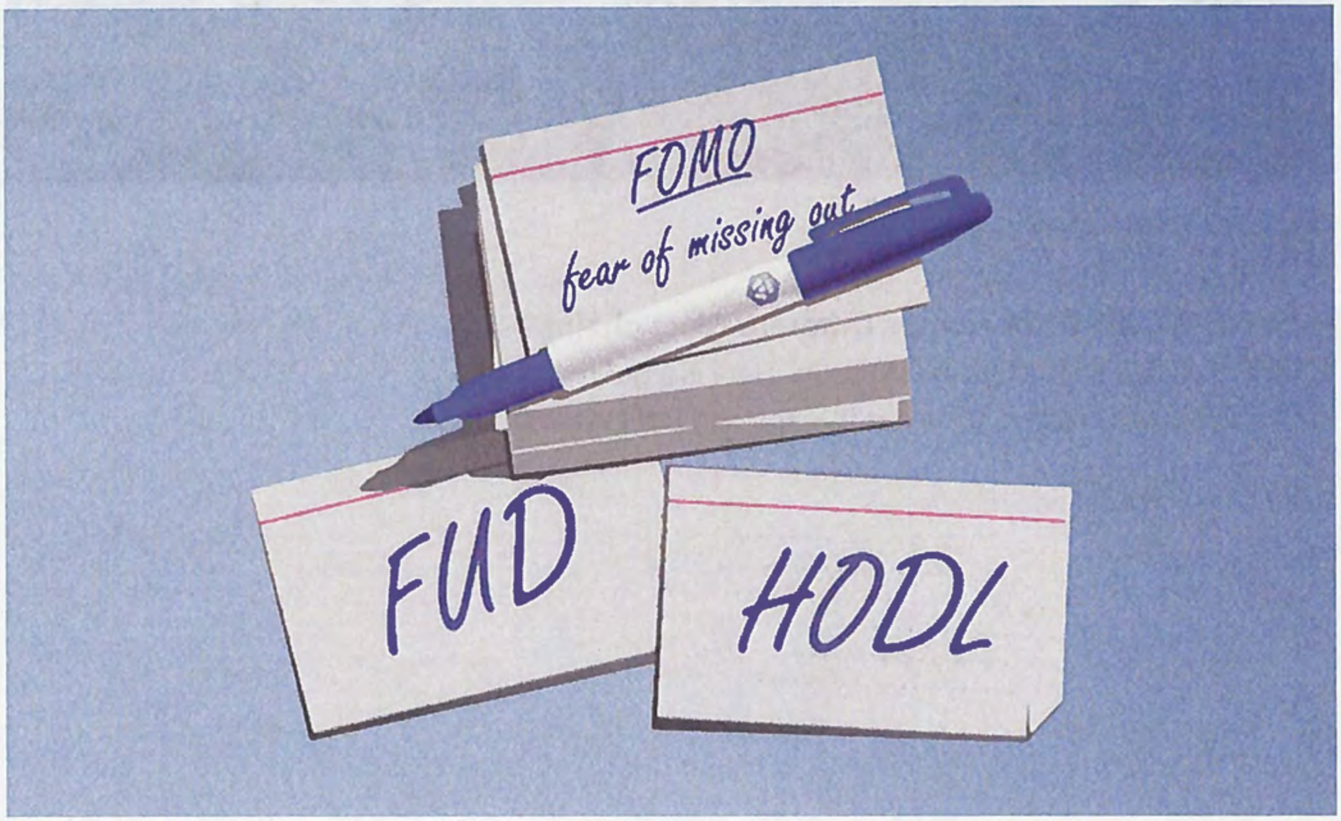
JUL 11, 2022

What Are Governance Tokens?

Governance tokens give holders a voice in crypto and blockchain projects. Learn how it works in detail.

[redacted]

Read More - (undefined minute)



UNIVERSITY / COINS & TOKENS

AUG 31, 2022

Crypto Slang: 28 Terms You Should Know

The crypto world is full of technical jargon, slang, and acronyms. Here's everything to know so you can join the conversation.

Read More - (undefined minute)

[redacted]

UNIVERSITY / COINS & TOKENS

SEP 20, 2022

How Does CRO Work Across Different Blockchains?

CRO is available on multiple blockchains. Here is a guide on using and bridging the different tokens in the App and DeFi Wallet.

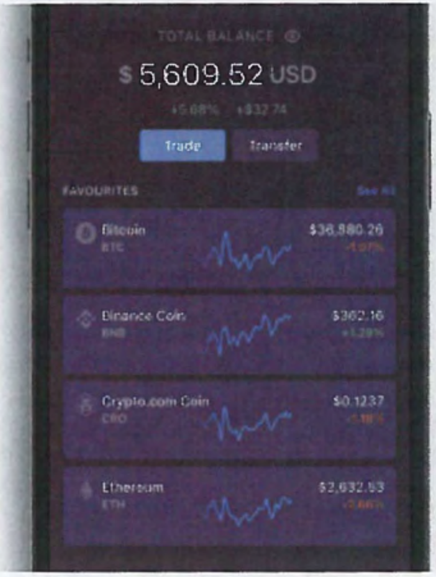
Read More - (undefined minute)

Ready to start your crypto journey?

Get your step-by-step guide to setting up an account with Crypto.com

Your Email Get Started

By clicking the Get Started button you acknowledge having read the Privacy Notice of Crypto.com where we explain how we use and protect your personal data.



TOTAL BALANCE

$ 5,609.52 USD

+5.68% +$32.74

Trade Transfer

FAVOURITES See All

Bitcoin BTC $36,880.26 -1.07%

Binance Coin BNB $362.16 +1.28%

Crypto.com Coin CRO $0.1237 -1.18%

Ethereum ETH $2,632.53 -2.66%

App

Cards

Commerce

Blockchain

DeFi

Resources

Learn

Company

NFT NEW

The purpose of this website is solely to display information regarding the products and services available on the Crypto.com App. It is not intended to offer access to any of such products and services. You may obtain access to such products and services on the Crypto.com App.

Please note that the availability of the products and services on the Crypto.com App is subject to jurisdictional limitations. Crypto.com may not offer certain products, features and/or services on the Crypto.com App in certain jurisdictions due to potential or actual regulatory restrictions.

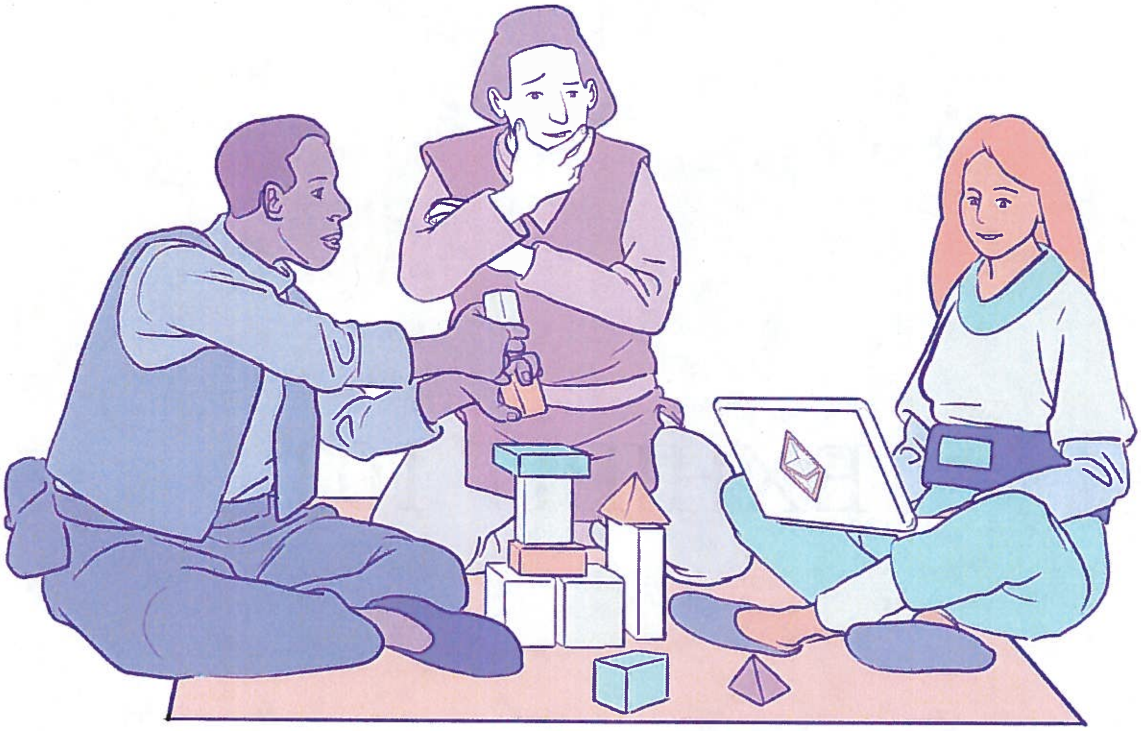
Crypto.com services are provided by Foris DAX, Inc. and its affiliates (NMLS ID 1966158).

Privacy Notice Do Not Sell My Personal Information Legal Status Cookie Preferences

EXHIBIT 103

Intro to Ethereum

ethereum.org/en/developers/docs/intro-to-ethereum/



Last edit: , Invalid DateTime

Edit page

On this page

What is a blockchain?

A blockchain is a public database that is updated and shared across many computers in a network.

"Block" refers to data and state being stored in consecutive groups known as "blocks". If you send ETH to someone else, the transaction data needs to be added to a block to be successful.

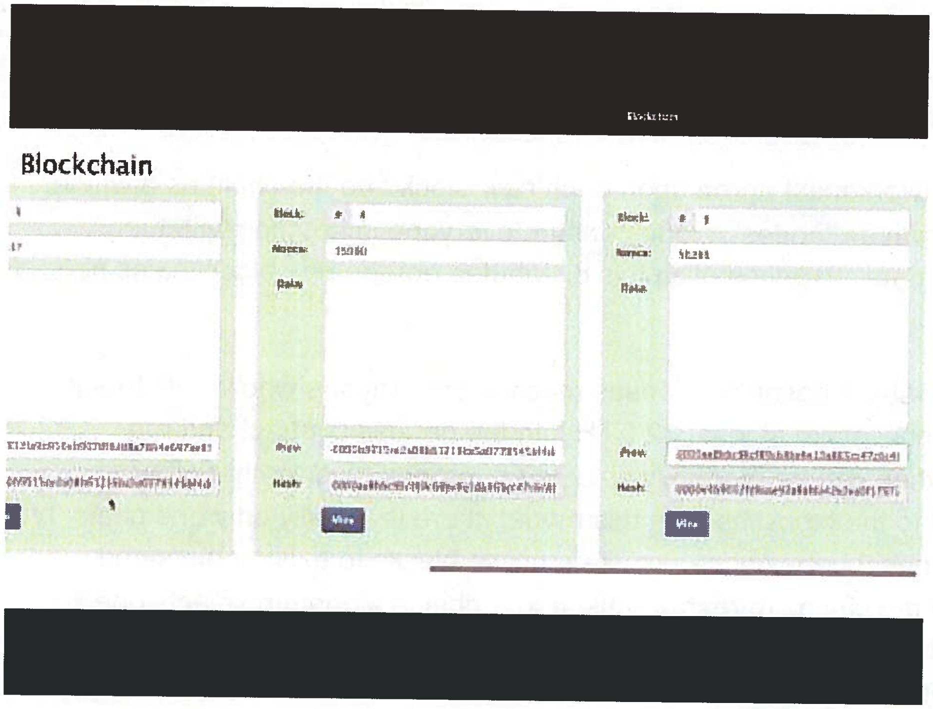
"Chain" refers to the fact that each block cryptographically references its parent. In other words, blocks get chained together. The data in a block cannot change without changing all subsequent blocks, which would require the consensus of the entire network.

Every computer in the network must agree upon each new block and the chain as a whole. These computers are known as "nodes". Nodes ensure everyone interacting with the blockchain has the same data. To accomplish this distributed agreement, blockchains need a consensus mechanism.

Ethereum uses a proof-of-stake-based consensus mechanism. Anyone who wants to add new blocks to the chain must stake at least 32 ETH into the deposit contract and run validator software. They then can be randomly selected to propose blocks that other validators check and add to the blockchain. In this model, there is usually only one chain, but network latency and dishonest behavior can cause multiple blocks to exist at the same position near the head of the chain. To resolve this, a fork-choice algorithm selects one canonical set of blocks. The blocks selected are the ones that form the heaviest possible chain, where 'heavy' refers to the number of validators that have endorsed the blocks (weighted by the ETH they have staked). There is a system of rewards and penalties that strongly incentivize participants to be honest and online as much as possible.

If you want to see how blockchain hashes data and then the previous block references all the past blocks, be sure to check out this demo by Anders Brownworth and watch the accompanying video below.

Watch Anders explain hashes in blockchains:



Watch Video At: https://youtu.be/_160oMzbIY8

What is Ethereum?

Ethereum is a blockchain with a computer embedded in it. It is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.

In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on. Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer. Additionally, any participant can broadcast a request for this computer to perform arbitrary computation. Whenever such a request is broadcast, other participants on the network verify, validate, and carry out ("execute") the computation. This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.

Requests for computation are called transaction requests; the record of all transactions and the EVM's present state gets stored on the blockchain, which in turn is stored and agreed upon by all nodes.

Cryptographic mechanisms ensure that once transactions are verified as valid and added to the blockchain, they can't be tampered with later. The same mechanisms also ensure that all transactions are signed and executed with appropriate "permissions" (no one should be able to send digital assets from Alice's account, except for Alice herself).

What is ether?

Ether (ETH) is the native cryptocurrency of Ethereum. The purpose of ETH is to allow for a market for computation. Such a market provides an economic incentive for participants to verify and execute transaction requests and provide computational resources to the network.

Any participant who broadcasts a transaction request must also offer some amount of ETH to the network as a bounty. The network will award this bounty to whoever eventually does the work of verifying the transaction, executing it, committing it to the blockchain, and broadcasting it to the network.

The amount of ETH paid corresponds to the time required to do the computation. These bounties also prevent malicious participants from intentionally clogging the network by requesting the execution of infinite computation or other resource-intensive scripts, as these participants must pay for computation time.

ETH is also used to provide crypto-economic security to the network in three main ways: 1) it is used as a means to reward validators who propose blocks or call out dishonest behavior by other validators; 2) It is staked by validators, acting as collateral against dishonest behavior—if validators attempt to misbehave their ETH can be destroyed; 3) it is used to weight 'votes' for newly proposed blocks, feeding into the fork-choice part of the consensus mechanism.

What are smart contracts?

In practice, participants don't write new code every time they want to request a computation on the EVM. Rather, application developers upload programs (reusable snippets of code) into EVM state, and users make requests to execute these code snippets with varying parameters. We call the programs uploaded to and executed by the network smart contracts.

At a very basic level, you can think of a smart contract like a sort of vending machine: a script that, when called with certain parameters, performs some actions or computation if certain conditions are satisfied. For example, a simple vendor smart contract could create and assign ownership of a digital asset if the caller sends ETH to a specific recipient.

Any developer can create a smart contract and make it public to the network, using the blockchain as its data layer, for a fee paid to the network. Any user can then call the smart contract to execute its code, again for a fee paid to the network.

Thus, with smart contracts, developers can build and deploy arbitrarily complex user-facing apps and services such as: marketplaces, financial instruments, games, etc.

Terminology

Blockchain

The sequence of all blocks that have been committed to the Ethereum network in the history of the network. So named because each block contains a reference to the previous block, which helps us maintain an ordering over all blocks (and thus over the precise history).

ETH

Ether (ETH) is the native cryptocurrency of Ethereum. Users pay ETH to other users to have their code execution requests fulfilled.

More on ETH

EVM

The Ethereum Virtual Machine is the global virtual computer whose state every participant on the Ethereum network stores and agrees on. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.

More on the EVM

Nodes

The real-life machines which are storing the EVM state. Nodes communicate with each other to propagate information about the EVM state and new state changes. Any user can also request the execution of code by broadcasting a code execution request from a node. The Ethereum network itself is the aggregate of all Ethereum nodes and their communications.

More on nodes

Accounts

Where ETH is stored. Users can initialize accounts, deposit ETH into the accounts, and transfer ETH from their accounts to other users. Accounts and account balances are stored in a big table in the EVM; they are a part of the overall EVM state.

More on accounts

Transactions

A "transaction request" is the formal term for a request for code execution on the EVM, and a "transaction" is a fulfilled transaction request and the associated change in the EVM state. Any user can broadcast a transaction request to the network from a node. For the transaction request to affect the agreed-upon EVM state, it must be validated, executed, and "committed to the network" by another node. Execution of any code causes a state change in the EVM; upon commitment, this state change is broadcast to all nodes in the network. Some examples of transactions:

- Send X ETH from my account to Alice's account.
- Publish some smart contract code into EVM state.
- Execute the code of the smart contract at address X in the EVM, with arguments Y.

More on transactions

Blocks

The volume of transactions is very high, so transactions are "committed" in batches, or blocks. Blocks generally contain dozens to hundreds of transactions.

More on blocks

Smart contracts

A reusable snippet of code (a program) which a developer publishes into EVM state. Anyone can request that the smart contract code be executed by making a transaction request. Because developers can write arbitrary executable applications into the EVM (games, marketplaces, financial instruments, etc.) by publishing smart contracts, these are often also called dapps, or Decentralized Apps.

More on smart contracts

Further reading

Know of a community resource that helped you? Edit this page and add it!

Related tutorials

A developer's guide to Ethereum, part 1 – *A very beginner-friendly exploration of Ethereum using Python and web3.py*

Back to top ↑

Was this article helpful?

Next

Intro to Ether

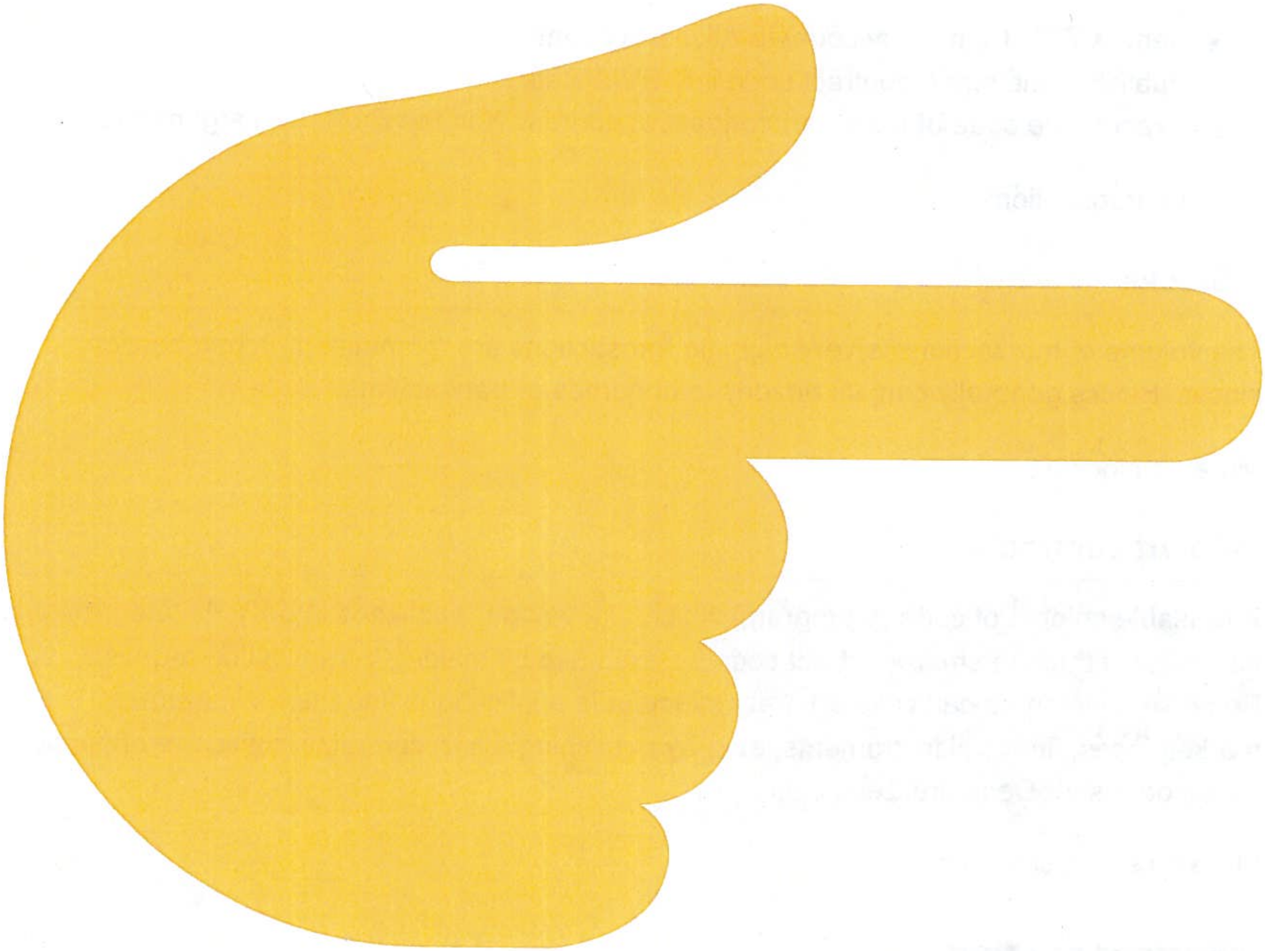
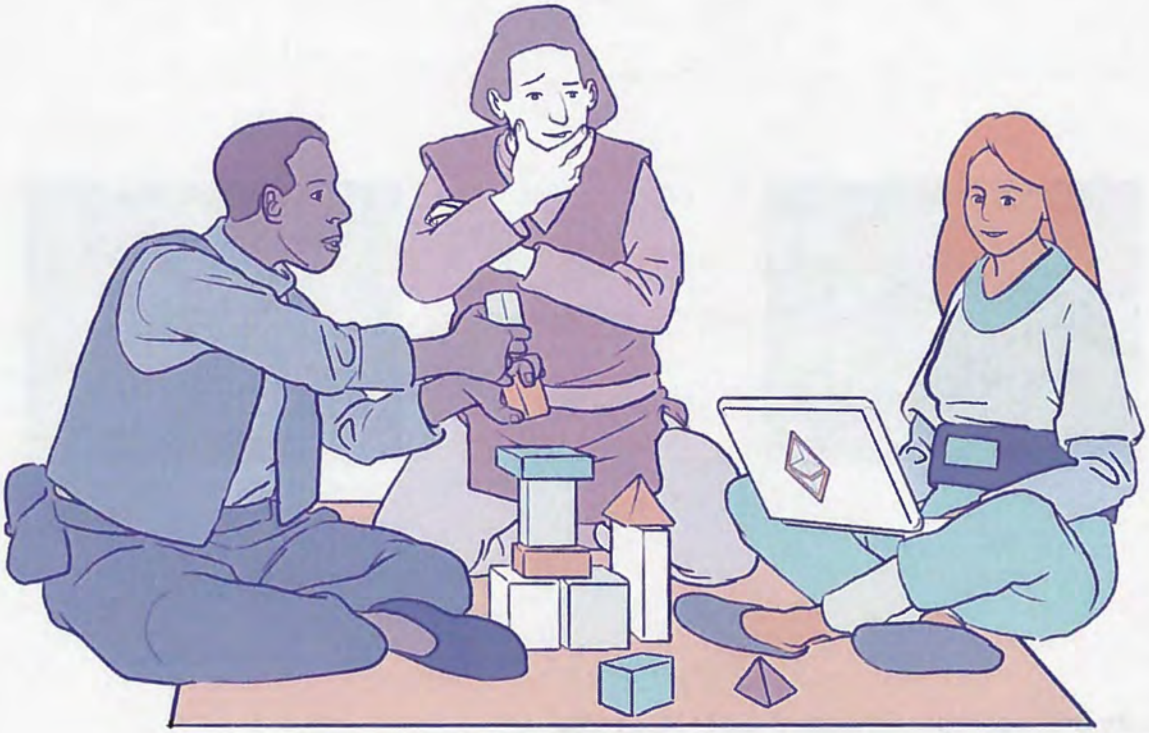


EXHIBIT 107

[redacted]

Transactions

ethereum.org/en/developers/docs/transactions/



Last edit: , Invalid DateTime

Edit page

On this page

Transactions are cryptographically signed instructions from accounts. An account will initiate a transaction to update the state of the Ethereum network. The simplest transaction is transferring ETH from one account to another.

Prerequisites

To help you better understand this page, we recommend you first read Accounts and our introduction to Ethereum.

What's a transaction?

An Ethereum transaction refers to an action initiated by an externally-owned account, in other words an account managed by a human, not a contract. For example, if Bob sends Alice 1 ETH, Bob's account must be debited and Alice's must be credited. This state-changing action takes place within a transaction.

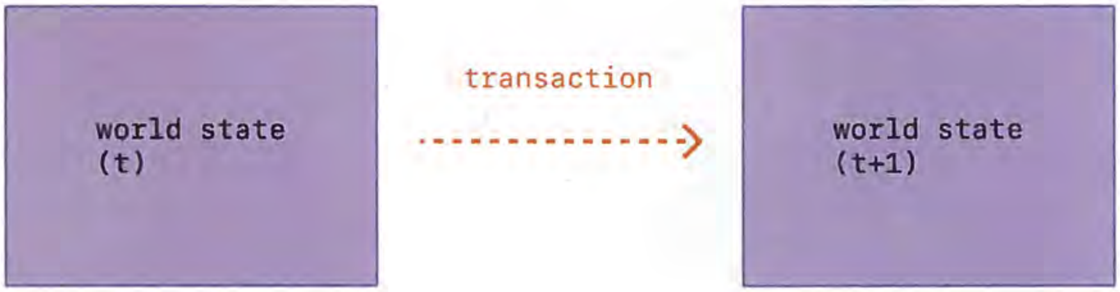


Diagram adapted from Ethereum EVM illustrated

Transactions, which change the state of the EVM, need to be broadcast to the whole network. Any node can broadcast a request for a transaction to be executed on the EVM; after this happens, a validator will execute the transaction and propagate the resulting state change to the rest of the network.

Transactions require a fee and must be included in a validated block. To make this overview simpler we'll cover gas fees and validation elsewhere.

A submitted transaction includes the following information:

- `recipient` – the receiving address (if an externally-owned account, the transaction will transfer value. If a contract account, the transaction will execute the contract code)
- `signature` – the identifier of the sender. This is generated when the sender's private key signs the transaction and confirms the sender has authorized this transaction
- `nonce` - a sequencially incrementing counter which indicate the transaction number from the account
- `value` – amount of ETH to transfer from sender to recipient (in WEI, a denomination of ETH)
- `data` – optional field to include arbitrary data

- `gasLimit` – the maximum amount of gas units that can be consumed by the transaction. Units of gas represent computational steps
- `maxPriorityFeePerGas` - the maximum amount of gas to be included as a tip to the validator
- `maxFeePerGas` - the maximum amount of gas willing to be paid for the transaction (inclusive of `baseFeePerGas` and `maxPriorityFeePerGas`)

Gas is a reference to the computation required to process the transaction by a validator. Users have to pay a fee for this computation. The `gasLimit` , and `maxPriorityFeePerGas` determine the maximum transaction fee paid to the validator. More on Gas.

The transaction object will look a little like this:

```
{
from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
gasLimit: "21000",
maxFeePerGas: "300",
maxPriorityFeePerGas: "10",
nonce: "0",
value: "10000000000"
}

```

Show all

Copy

But a transaction object needs to be signed using the sender's private key. This proves that the transaction could only have come from the sender and was not sent fraudulently.

An Ethereum client like Geth will handle this signing process.

Example JSON-RPC call:

```
{
  "id": 2,
  "jsonrpc": "2.0",
  "method": "account_signTransaction",
  "params": [
    {
      "from": "0x1923f626bb8dc025849e00f99c25fe2b2f7fb0db",
      "gas": "0x55555",
      "maxFeePerGas": "0x1234",
      "maxPriorityFeePerGas": "0x1234",
      "input": "0xabcd",
      "nonce": "0x0",
      "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
      "value": "0x1234"
    }
  ]
}

```

Show all

Copy

Example response:

```
{
"jsonrpc": "2.0",
"id": 2,
"result": {
  "raw":
"0xf88380018203339407a565b7ed7d7a678680a4c162885bedbb695fe080a44401a6e40000000000000000

  "tx": {
    "nonce": "0x0",
    "maxFeePerGas": "0x1234",
    "maxPriorityFeePerGas": "0x1234",
     "gas": "0x55555",
     "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
     "value": "0x1234",
     "input": "0xabcd",
     "v": "0x26",
     "r": "0x223a7c9bcf5531c99be5ea7082183816eb20cfe0bbc322e97cc5c7f71ab8b20e",
     "s": "0x2aadee6b34b45bb15bc42d9c09de4a6754e7000908da72d48cc7704971491663",
     "hash": "0xeba2df809e7a612a0a0d444ccfa5c839624bdc00dd29e3340d46df3870f8a30e"
    }
  }
}

```

Show all

Copy

- the `raw` is the signed transaction in Recursive Length Prefix (RLP) encoded form
- the `tx` is the signed transaction in JSON form

With the signature hash, the transaction can be cryptographically proven that it came from the sender and submitted to the network.

The data field

The vast majority of transactions access a contract from an externally-owned account. Most contracts are written in Solidity and interpret their data field in accordance with the application binary interface (ABI)?.

The first four bytes specify which function to call, using the hash of the function's name and arguments. You can sometimes identify the function from the selector using this database.

The rest of the calldata is the arguments, encoded as specified in the ABI specs.

For example, lets look at this transaction. Use **Click to see More** to see the calldata.

The function selector is `0xa9059cbb` . There are several known functions with this signature. In this case the contract source code has been uploaded to Etherscan, so we know the function is `transfer(address,uint256)` .

The rest of the data is:

```
1000000000000000000000000004f6742badb049791cd9a37ea913f2bac38d01279

200000000000000000000000000000000000000000000000000000000003b0559f4

3
```

According to the ABI specifications, integer values (such as addresses, which are 20-byte integers) appear in the ABI as 32-byte words, padded with zeros in the front. So we know that the `to` address is `4f6742badb049791cd9a37ea913f2bac38d01279`. The `value` is 0x3b0559f4 = 990206452.

Types of transactions

On Ethereum there are a few different types of transactions:

- Regular transactions: a transaction from one account to another.
- Contract deployment transactions: a transaction without a 'to' address, where the data field is used for the contract code.
- Execution of a contract: a transaction that interacts with a deployed smart contract. In this case, 'to' address is the smart contract address.

On gas

As mentioned, transactions cost gas to execute. Simple transfer transactions require 21000 units of Gas.

So for Bob to send Alice 1 ETH at a `baseFeePerGas` of 190 gwei and `maxPriorityFeePerGas` of 10 gwei, Bob will need to pay the following fee:

```
(190 + 10) * 21000 = 4,200,000 gwei

--or--

0.0042 ETH

```

Bob's account will be debited **-1.0042 ETH** (1 ETH for Alice + 0.0042 ETH in gas fees)

Alice's account will be credited **+1.0 ETH**

The base fee will be burned **-0.00399 ETH**

Validator keeps the tip **+0.000210 ETH**

Gas is required for any smart contract interaction too.

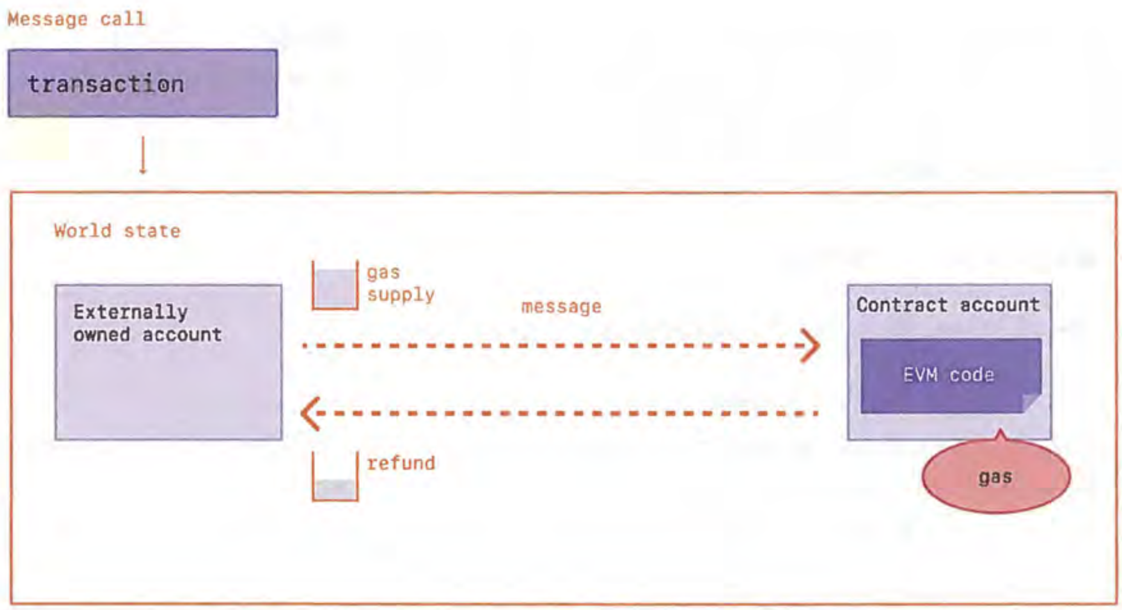


Diagram adapted from Ethereum EVM illustrated

Any gas not used in a transaction is refunded to the user account.

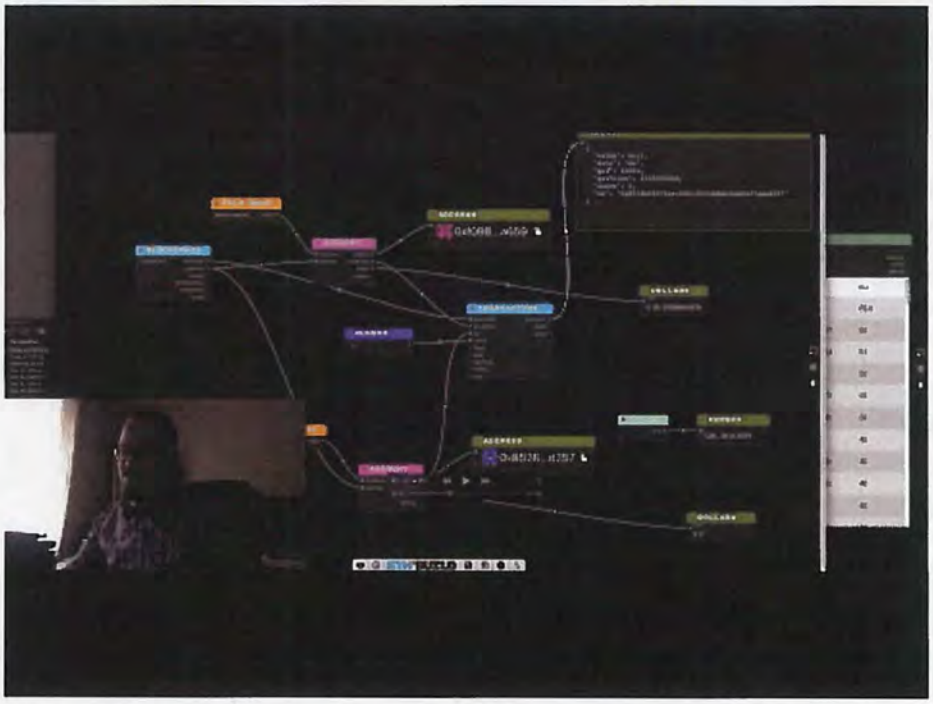
Transaction lifecycle

Once the transaction has been submitted the following happens:

1. Once you send a transaction, cryptography generates a transaction hash: `0x97d99bc7729211111a21b12c933c949d4f31684f1d6954ff477d0477538ff017`
2. The transaction is then broadcast to the network and included in a pool with lots of other transactions.
3. A validator must pick your transaction and include it in a block in order to verify the transaction and consider it "successful".
4. As time passes the block containing your transaction will be upgraded to "justified" then "finalized". These upgrades make it much more certain that your transaction was successful and will never be altered. Once a block is "finalized" it could only ever be changed by an attack that would cost many billions of dollars.

A visual demo

Watch Austin walk you through transactions, gas, and mining.



Watch Video At: https://youtu.be/er-0ihqFQB0

Typed Transaction Envelope

Ethereum originally had one format for transactions. Each transaction contained a nonce, gas price, gas limit, to address, value, data, v, r, and s. These fields are RLP-encoded, to look something like this:

```
RLP([nonce, gasPrice, gasLimit, to, value, data, v, r, s])
```

Ethereum has evolved to support multiple types of transactions to allow for new features such as access lists and EIP-1559 to be implemented without affecting legacy transaction formats.

EIP-2718: Typed Transaction Envelope defines a transaction type that is an envelope for future transaction types.

EIP-2718 is a new generalised envelope for typed transactions. In the new standard, transactions are interpreted as:

```
TransactionType || TransactionPayload
```

Where the fields are defined as:

- `TransactionType` - a number between 0 and 0x7f, for a total of 128 possible transaction types.
- `TransactionPayload` - an arbitrary byte array defined by the transaction type.

Further reading

EIP-2718: Typed Transaction Envelope

Know of a community resource that helped you? Edit this page and add it!

Related topics

Back to top ↑

Was this article helpful?

👈

Previous

Accounts
Next

Blocks

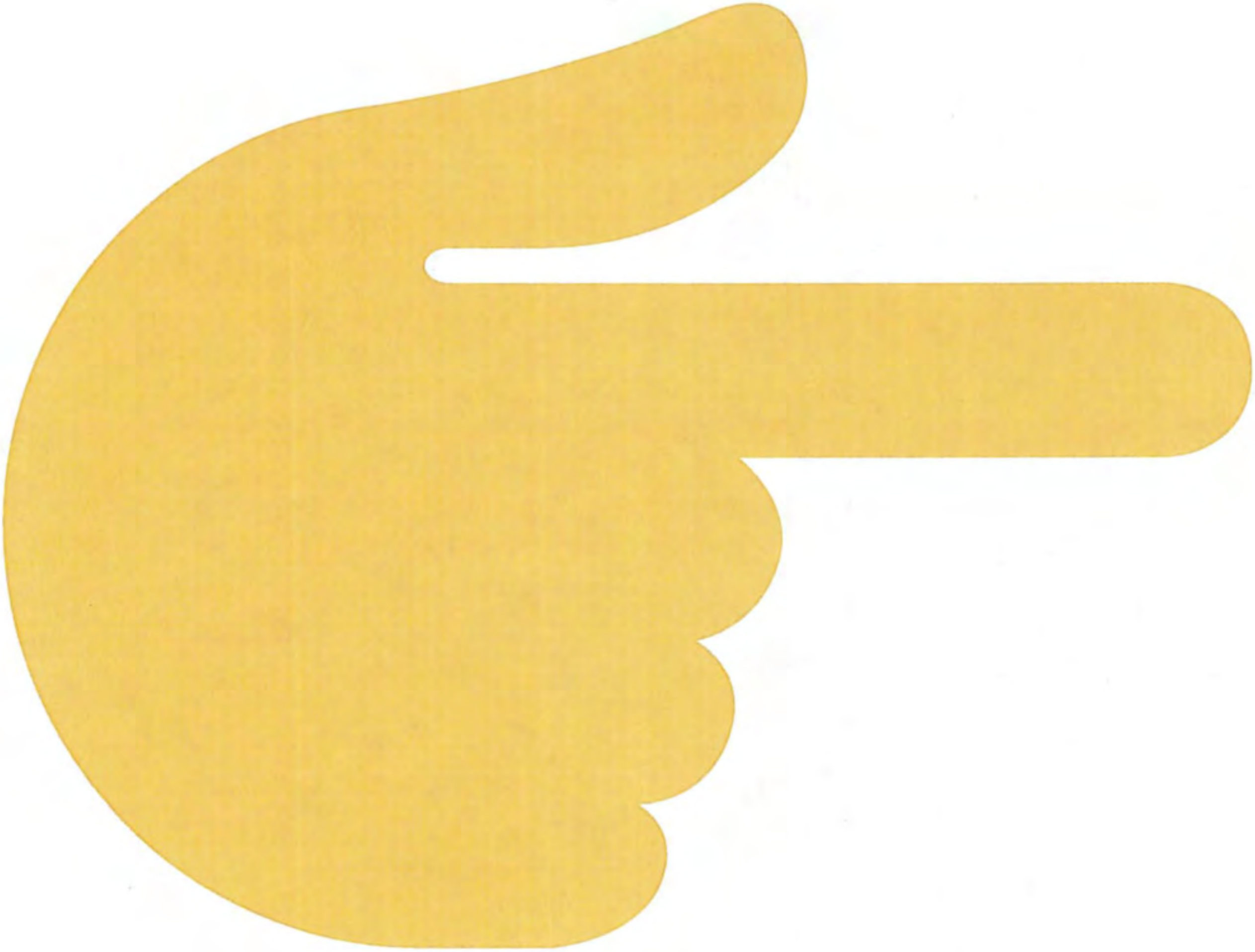


EXHIBIT 108

[redacted]

What is Blockchain Analysis? - Blog

certik.com/resources/blog/what-is-blockchain-analysis

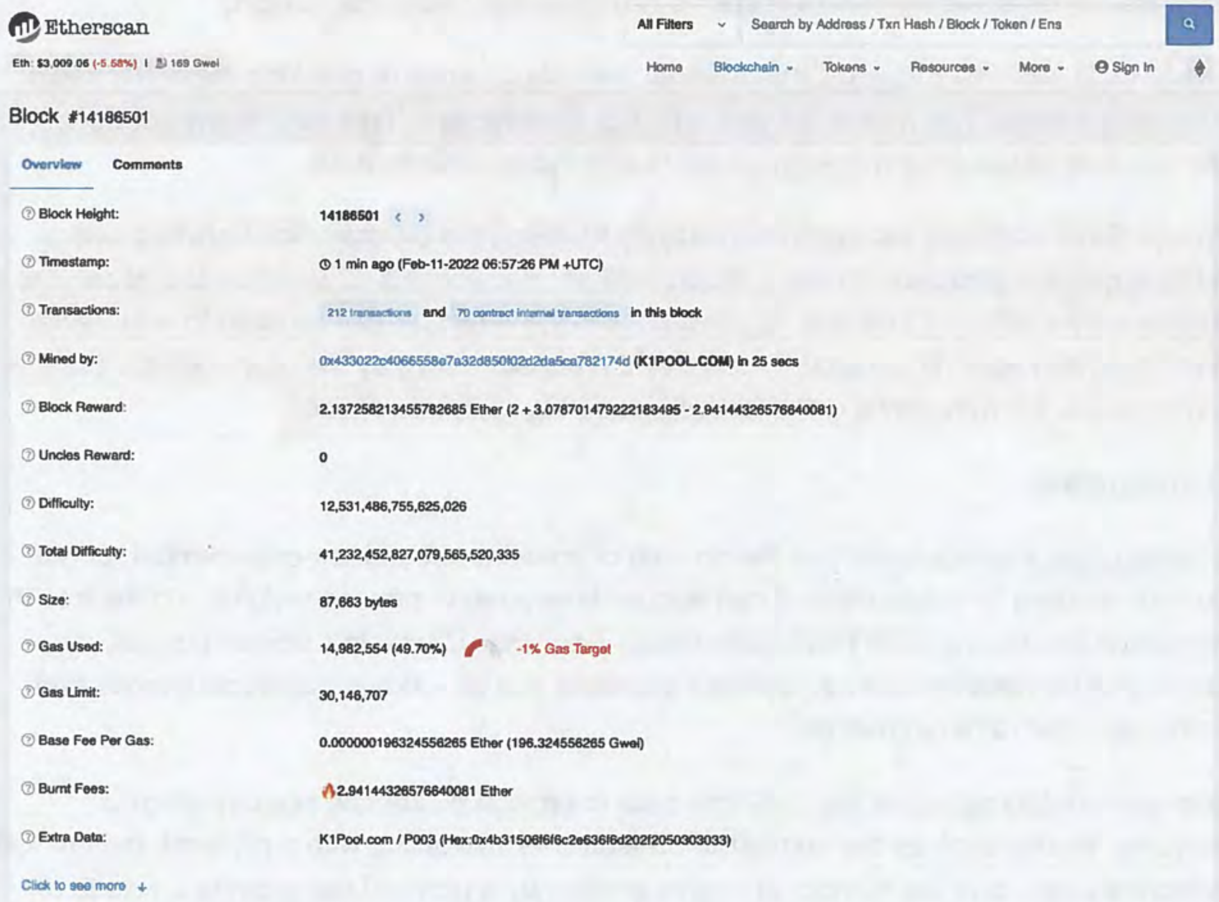
← Back to all stories

Blockchain analysis is the process of inspecting, cataloging, and interpreting the data that blockchains produce in order to gain actionable insights.

A public blockchain, such as Bitcoin or Ethereum, is essentially a database of accounts and their respective balances. Each new block in the chain updates the previous state of the database. With the Ethereum blockchain currently growing at a rate of more than 100GB per year, this represents an enormous amount of information added to the permanent history of the chain. And keep in mind that all this data is in text format, and 100GB of text is very different to 100GB of images or video.

This information is free for anyone and everyone to browse, but it exists in a raw, unprocessed state.

Tools such as blockchain explorers perform the fundamental tasks of organizing this data.



Block #14186501

Block Height:	14186501 ‹ ›
Timestamp:	1 min ago (Feb-11-2022 06:57:26 PM +UTC)
Transactions:	212 transactions and 70 contract internal transactions in this block
Mined by:	0x433022c4066558e7a32d850f02d2da5ca782174d (K1POOL.COM) in 25 secs
Block Reward:	2.137258213455782685 Ether (2 + 3.078701479222183495 - 2.94144326576640081)
Uncles Reward:	0
Difficulty:	12,531,486,755,625,026
Total Difficulty:	41,232,452,827,079,565,520,335
Size:	87,663 bytes
Gas Used:	14,982,554 (49.70%) -1% Gas Target
Gas Limit:	30,146,707
Base Fee Per Gas:	0.000000196324556265 Ether (196.324556265 Gwei)
Burnt Fees:	2.94144326576640081 Ether
Extra Data:	K1Pool.com / P003 (Hex:0x4b31506f6f6c2e636f6d202f2050303033)

An Ethereum block's data on Etherscan

Block height, total difficulty, hashes, parent hashes, state roots, nonces… it's hard for anyone who doesn't make a career of blockchain analysis to differentiate between the useful and irrelevant information that blockchains create.

What Can You Do With Blockchain Analysis?

Before we get into how you can perform blockchain analysis, let's discuss why you might want to.

One of the primary purposes of blockchain analysis is to trace the flow of funds between addresses. This may be to follow the proceeds of an exploit, or to establish a transaction chain linking two or more wallets. Law enforcement agencies, such as the U.S. Department of Justice's newly launched National Cryptocurrency Enforcement Team (NCET), make extensive use of blockchain analysis when conducting anti-money laundering and cybersecurity operations. While no laws exist at the federal level regarding the admissibility of blockchain data in court cases, states including Arizona have passed laws confirming the legal validity of blockchain records. The immutability of blockchains makes the technology well-suited to establishing historical claims and chains of strong correlation.

Privacy tools such as Tornado Cash exist for the sole purpose of breaking these traceable transaction chains. This makes it a powerful tool for anyone to take back some control of their online financial privacy, everyday users and cybercriminals alike.

Tornado Cash improves transaction privacy by breaking the on-chain link between source and destination addresses. It uses a smart contract that accepts ETH deposits that can be withdrawn by a different address. To preserve privacy a relayer can be used to withdraw to an address with no ETH balance. Whenever ETH is withdrawn by the new address, there is no way to link the withdrawal to the deposit, ensuring complete privacy.

– Tornado Cash

But blockchain analysis is not just the domain of government and law enforcement, or the criminals seeking to evade them. It can also be leveraged to provide insights into the health and overall functioning of all blockchain-based platforms. Crypto is a unique industry, as actions and transactions are not reported quarterly, if at all – like in traditional finance and commerce – but rather in real-time.

Tools such as Skynet utilize this real-time data to provide actionable security insights. Analyzing metrics such as the number of transactions interacting with a protocol, the number of discrete users, and the number of events emitted by a protocol can provide a wealth of information that paints a specific picture of a platform's functioning over time. Individual traders and investors can make use of these tools to monitor platforms and projects in which they have invested.

DeFi users can also utilize several different platforms that monitor their wallets – or anyone else's wallet – and send an alert whenever a transaction is processed. This allows them to have real-time notifications about any activity on the addresses most important to them, whether it's final confirmation of a low-priority transaction broadcast to the network hours before, or the first attempt of a hacker to gain control of or drain funds from their account.

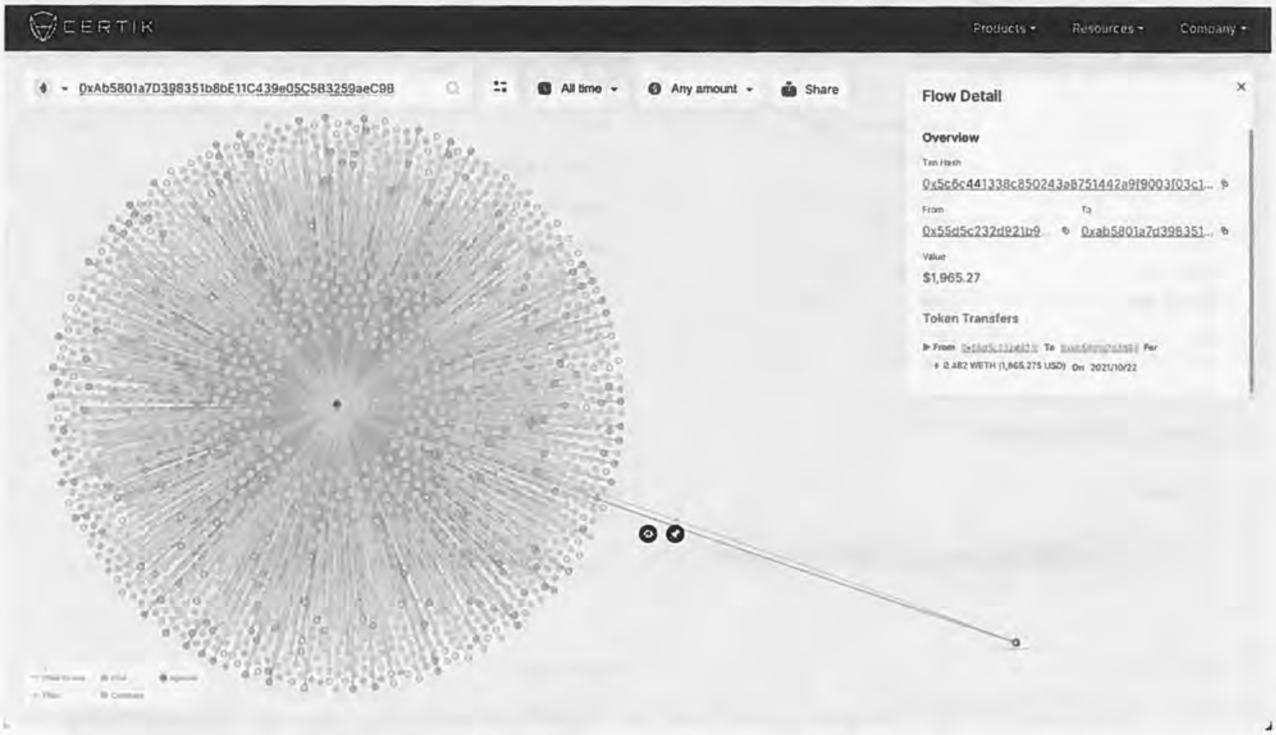
Blockchain analysis is a rich and continuously expanding field. Now that we've gone over what blockchain analysis is, let's take a look at the tools that are available to help you leverage it in your research and how exactly these tools work.

What Blockchain Analysis Tools are Available?

Etherscan is likely to be the tool that DeFi users are most familiar with. Many DeFi platforms provide a direct link to the website after every transaction, so the user can check on its confirmation status. Etherscan is an excellent tool for confirming transactions and getting a general overview of a particular wallet's holdings. Additionally, the fact that the site tags many well known wallets (e.g. Coinbase 1 Hot Wallet or Uniswap V3 Router) makes it relatively easy to see at a glance where your money is coming from and going.

Etherscan is great for raw data such as wallet balances and transaction history; the who, what, and how much. But if you're looking for second-level insights, you'll need to turn to a tool that aggregates and analyzes this raw data.

Skytrace is a blockchain analysis tool that greatly simplifies the process of tracing the flow of funds from one wallet to another. Skytrace visualizes a wallet's interactions with other addresses and has helpful tags for well-known protocols, such as Uniswap and Tornado Cash.



0xAb5801a7D398351b8bE11C439e05C5B3259aeC9B

Flow Detail

Overview

Txn Hash
0x5c6c441338c850243a8751442a9f9003f03c1...

From
0x55d5c232d921b9...

To
0xab5801a7d398351...

Value
$1,965.27

Token Transfers

▶ From [illegible] To [illegible] For
+ 0.482 WETH (1,965.275 USD) On 2021/10/22

Using Skytrace to visualize one of Vitalik Buterin's hundreds of transactions, in this case his sale of 6,757.307 HEX for 0.48 ETH on Uniswap

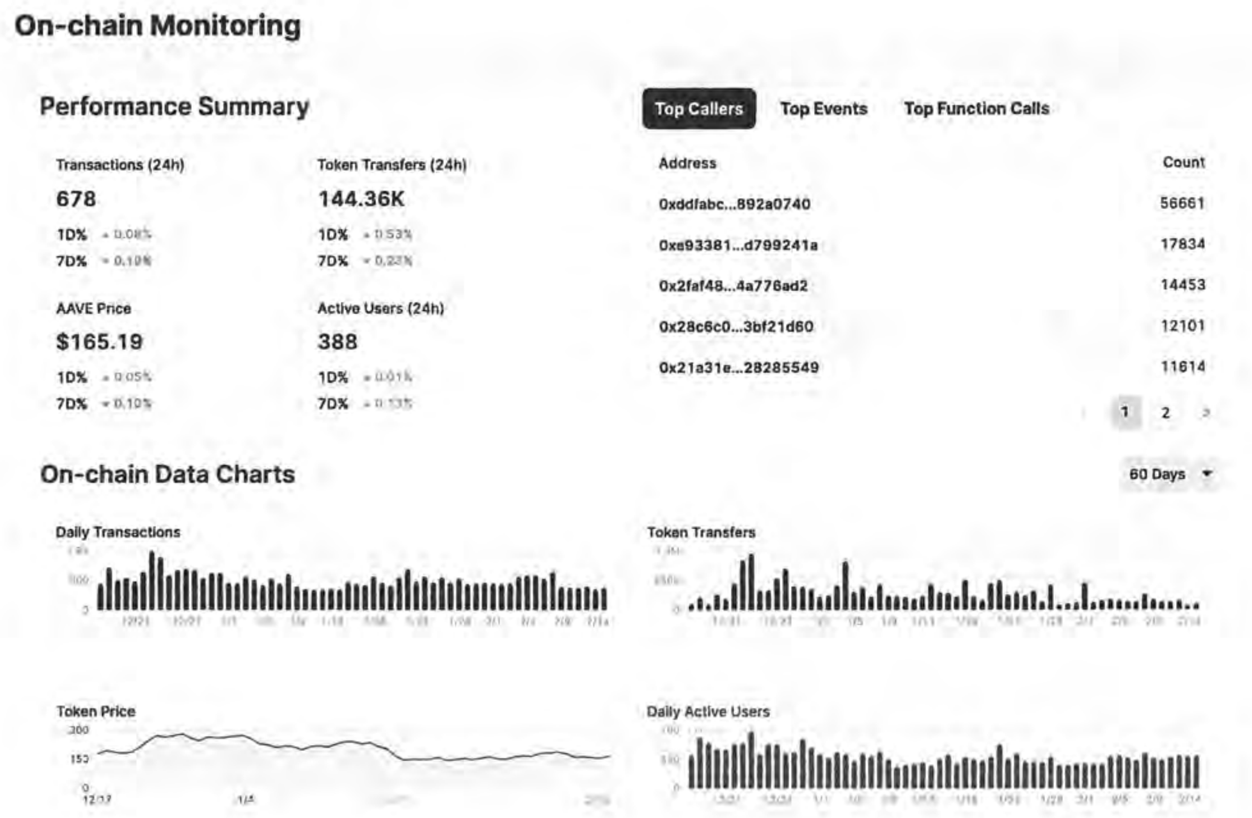
CertiK's Skynet combines six security primitives to arrive at a comprehensive score that reflects the effectiveness of a DeFi project's security measures. These six primitives are: social sentiment, on-chain monitoring, governance, market dynamics, safety analysis, and finally the Security Oracle. With the exception of the social sentiment metric, each of these primitives incorporate blockchain analysis.

Breaking down each of these primitives will provide illustrative case studies of how blockchain analysis works.

How Does Blockchain Analysis Work?

Blockchain analysis works by aggregating the massive amount of data that blockchains produce, and then filtering, modeling, or otherwise inspecting it in order to produce actionable insights. These insights could be anything from a transaction linking two wallets, an important wallet making moves before or after big announcements, or a gradual decay in the number of active users of a specific DeFi protocol.

That's the short answer. To really understand how blockchain analysis works, let's take a deep dive into a specific example of on-chain monitoring. Skynet is CertiK's security scoring tool that uses on-chain analysis to arrive at actionable security and data insights.



On-chain Monitoring

Performance Summary

Transactions (24h)
678
1D% ▲ 0.08%
7D% ▼ 0.10%

Token Transfers (24h)
144.36K
1D% ▲ 0.53%
7D% ▼ 0.23%

AAVE Price
$165.19
1D% ▲ 0.05%
7D% ▼ 0.10%

Active Users (24h)
388
1D% ▲ 0.01%
7D% ▲ 0.13%

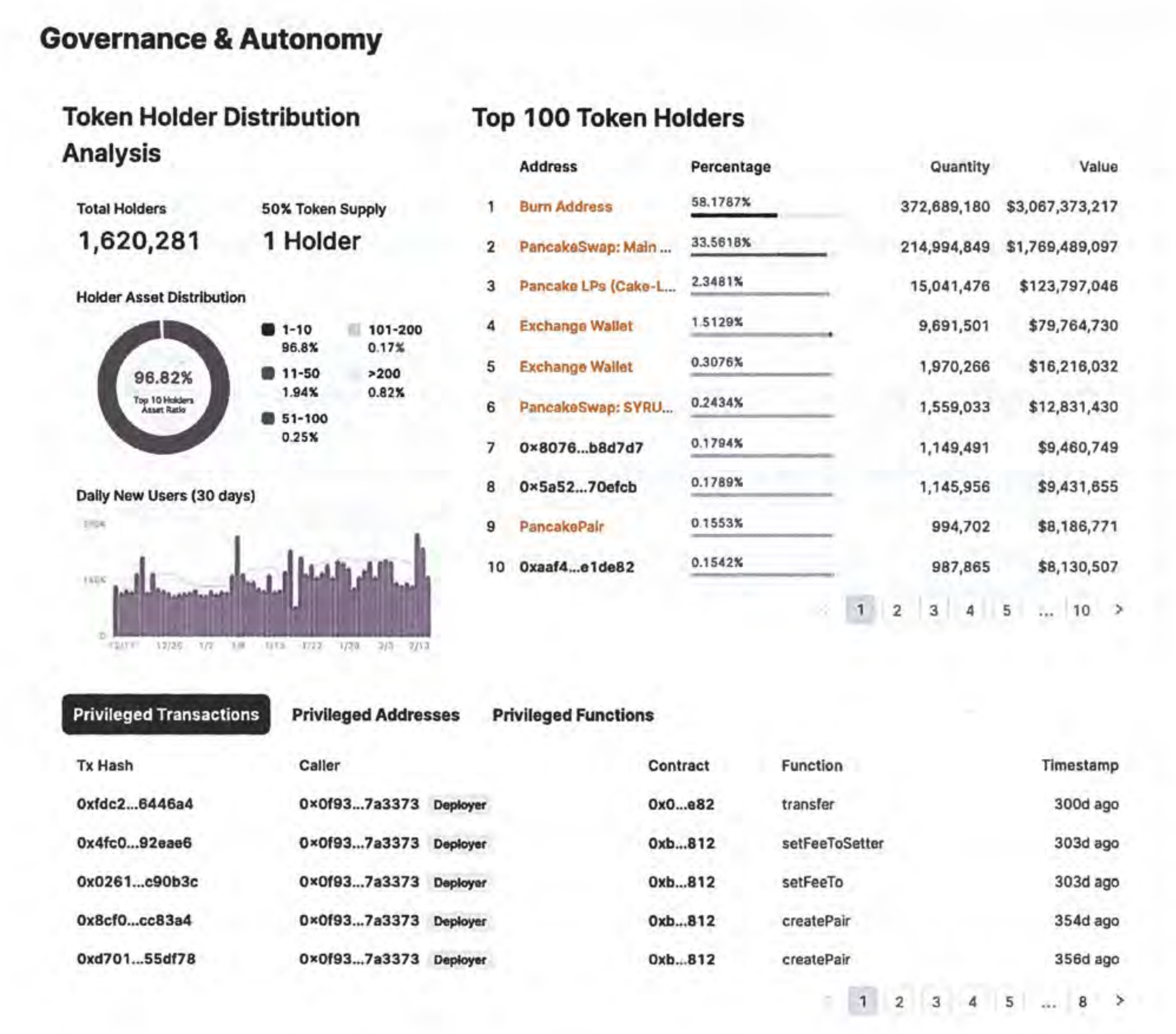
Address	Count
0xddfabc...892a0740	56661
0xe93381...d799241a	17834
0x2faf48...4a776ad2	14453
0x28c6c0...3bf21d60	12101
0x21a31e...28285549	11614

On-chain Data Charts

AAVE's entry on the Security Leaderboard. The tab shown is Skynet's on-chain monitoring analysis

The on-chain monitoring section of the Security Leaderboard gives a comprehensive overview of a project's activity. You can see the number of transactions interacting with the protocol over the last 24 hours, the number of transfers of the project's token, the number of active users, and the price of the token, plus all of these metrics plotted over a customizable period.

The next Skynet primitive is Governance. Decentralized governance is one of the most important factors that put the De in DeFi. Decentralized protocols have governance forums where users can propose, debate, and vote on ideas in an open, collaborative process. Any protocol that is governed in such a way is a DAO – a Decentralized Autonomous Organization. As you can imagine, when the power to make any and all decisions lies with a DAO, it's important for investors to pay close attention to the votes and actions it undertakes (if they're not getting directly involved themselves).



Governance & Autonomy

Token Holder Distribution Analysis

Total Holders
1,620,281

50% Token Supply
1 Holder

Holder Asset Distribution

Daily New Users (30 days)

Top 100 Token Holders

	Address	Percentage	Quantity	Value
1	Burn Address	58.1787%	372,689,180	$3,067,373,217
2	PancakeSwap: Main ...	33.5618%	214,994,849	$1,769,489,097
3	Pancake LPs (Cake-L...	2.3481%	15,041,476	$123,797,046
4	Exchange Wallet	1.5129%	9,691,501	$79,764,730
5	Exchange Wallet	0.3076%	1,970,266	$16,216,032
6	PancakeSwap: SYRU...	0.2434%	1,559,033	$12,831,430
7	0×8076...b8d7d7	0.1794%	1,149,491	$9,460,749
8	0×5a52...70efcb	0.1789%	1,145,956	$9,431,655
9	PancakePair	0.1553%	994,702	$8,186,771
10	0xaaf4...e1de82	0.1542%	987,865	$8,130,507

Privileged Transactions Privileged Addresses Privileged Functions

Tx Hash	Caller	Contract	Function	Timestamp
0xfdc2...6446a4	0×0f93...7a3373 Deployer	0x0...e82	transfer	300d ago
0x4fc0...92eae6	0×0f93...7a3373 Deployer	0xb...812	setFeeToSetter	303d ago
0x0261...c90b3c	0×0f93...7a3373 Deployer	0xb...812	setFeeTo	303d ago
0x8cf0...cc83a4	0×0f93...7a3373 Deployer	0xb...812	createPair	354d ago
0xd701...55df78	0×0f93...7a3373 Deployer	0xb...812	createPair	356d ago

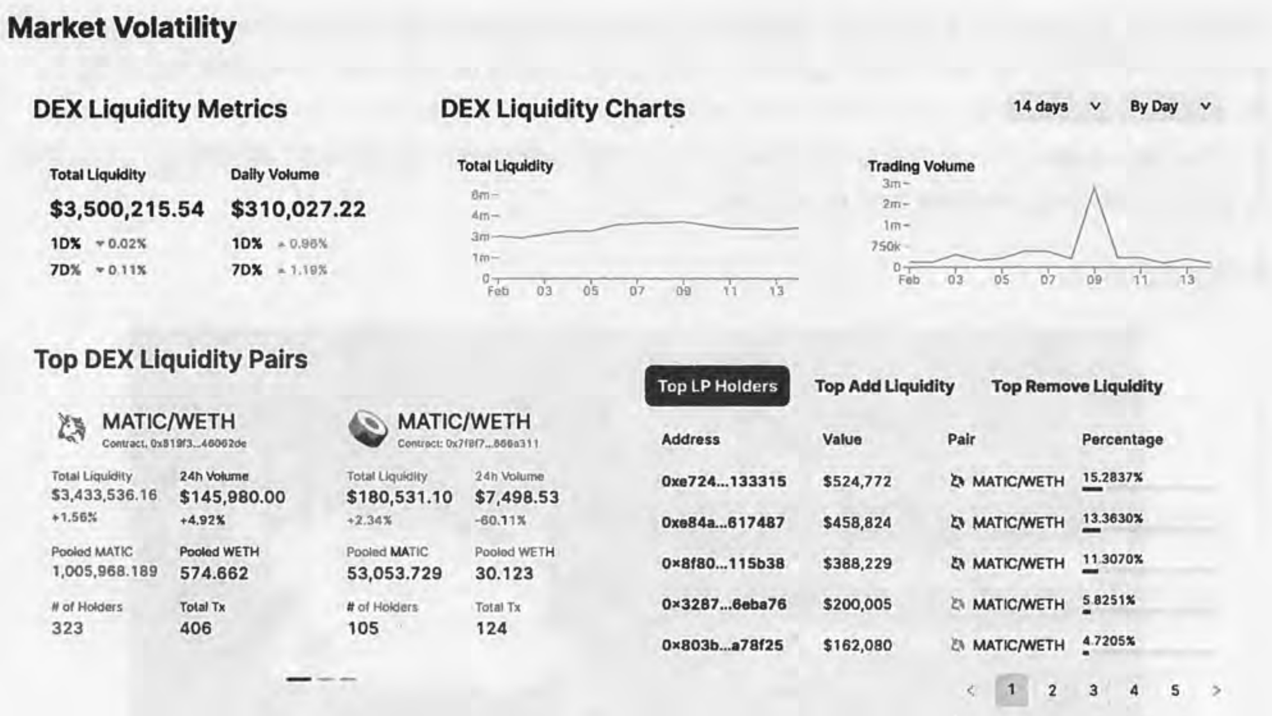
PancakeSwap's Skynet Governance Module

The Governance Score is an overall rating of the platform's decentralization, links to major crypto platforms, and the health of its DAO. The on-chain component of Skynet's Governance score is made up of the following metrics.

Privileged Transactions lists the number of privileged transactions in the last 72 hours. A privileged transaction is one initiated by an address that has power to modify a platform's smart contracts. A truly decentralized DeFi protocol should only be able to be updated or changed after its DAO has voted on and approved the changes. Recent Privileged Transactions is similar, but here we get a list of all privileged transactions, not just in the last 72 hours. It's a great way to see how often a platform's smart contracts are modified, by whom, and for what purpose.

The Privileged Addresses section lists all the addresses that have the power to initiate privileged transactions (as defined above). You can click on the address or contract to be taken to its listing on a block explorer – BSCScan in this case, since PancakeSwap runs on Binance Smart Chain.

Privileged Functions outlines the code functions that privileged addresses can invoke. In this case we've got burn, constructor, and mint. The burn function sends tokens to an address where they cannot be retrieved. The constructor function is called when initializing a contract. It sets the contract's variables to the correct state. Mint creates new tokens, often for liquidity mining rewards.



Market Volatility

DEX Liquidity Metrics

Total Liquidity
$3,500,215.54
1D% ▼0.02%
7D% ▼0.11%

Daily Volume
$310,027.22
1D% ▲0.96%
7D% ▲1.19%

DEX Liquidity Charts

14 days ∨ By Day ∨

Top DEX Liquidity Pairs

MATIC/WETH
Contract, 0x819f3...46062de

Total Liquidity $3,433,536.16 +1.56%	24h Volume $145,980.00 +4.92%
Pooled MATIC 1,005,968.189	Pooled WETH 574.662
# of Holders 323	Total Tx 406

MATIC/WETH
Contract: 0x7f8f7...866a311

Total Liquidity $180,531.10 +2.34%	24h Volume $7,498.53 -60.11%
Pooled MATIC 53,053.729	Pooled WETH 30.123
# of Holders 105	Total Tx 124

Top LP Holders Top Add Liquidity Top Remove Liquidity

Address	Value	Pair	Percentage
0xe724...133315	$524,772	MATIC/WETH	15.2837%
0xe84a...617487	$458,824	MATIC/WETH	13.3630%
0x8f80...115b38	$388,229	MATIC/WETH	11.3070%
0x3287...6eba76	$200,005	MATIC/WETH	5.8251%
0x803b...a78f25	$162,080	MATIC/WETH	4.7205%

< 1 2 3 4 5 >

Matic's entry on the Security Leaderboard

Decentralized exchanges (DEXs) function entirely on-chain, which means all the data they create is freely available. This is great news for anyone seeking to do research on decentralized market dynamics.

Skynet analyzes this data to provide actionable insights. At a single glance, you can see exactly which token pairs have the deepest liquidity, volume plotted over time, and the largest holders of liquidity provider (LP) tokens. All this data helps give an accurate overview of the health of a particular token's market dynamics. For example, if you see in the Top Remove Liquidity tab that all of the major LP Holders are suddenly withdrawing their positions, you may take that as a reason to do some further investigation.

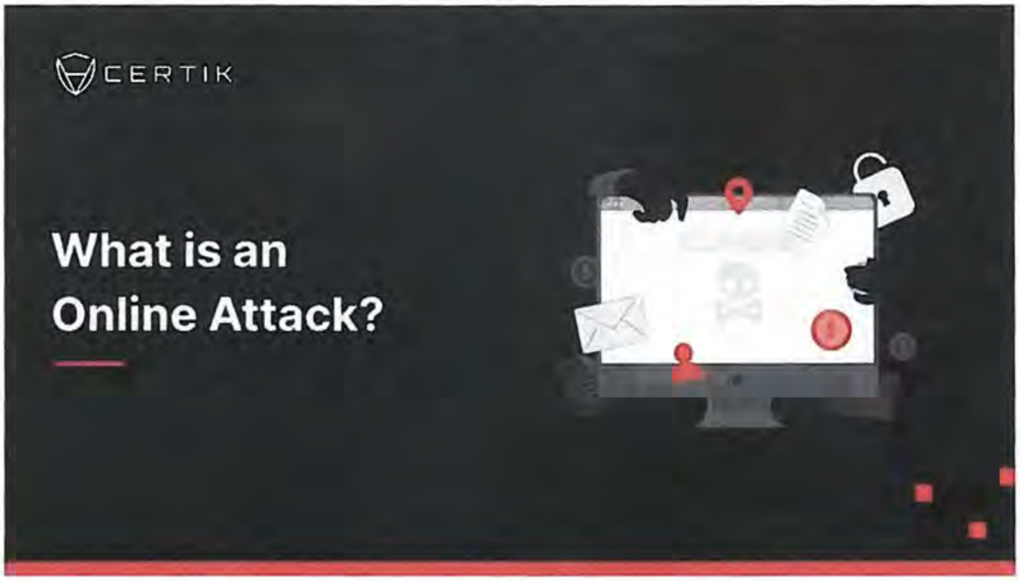
Blockchain Analysis – An Increasingly Powerful Tool

With the consistent growth of blockchain adoption and the data that users, platforms, and miners produce, blockchain analysis becomes more powerful every day. Larger datasets mean deeper insights. With the approaching Web3 economy focused on using blockchain technology to empower everyone left out of centralized Web2 platforms, blockchains are the newest frontier of data analysis.

Powerful tools exist to help all blockchain users gain insights into Web3 platforms. CertiK's Skynet is a security-focused resource that demystifies the complex technicalities of DeFi security, while Skytrace makes it easy to perform your own blockchain data analysis and map out interactions between wallets visually.

Blockchain analysis is a uniquely insightful way to understand the functioning and security of blockchain platforms. But it can also help you take control of your own personal security, by displaying exactly how much traceable information every transaction creates. As more and more value shifts onto public blockchains, it's worth remembering that all activity is out in the open, viewable by anyone and everyone.

Related Stories

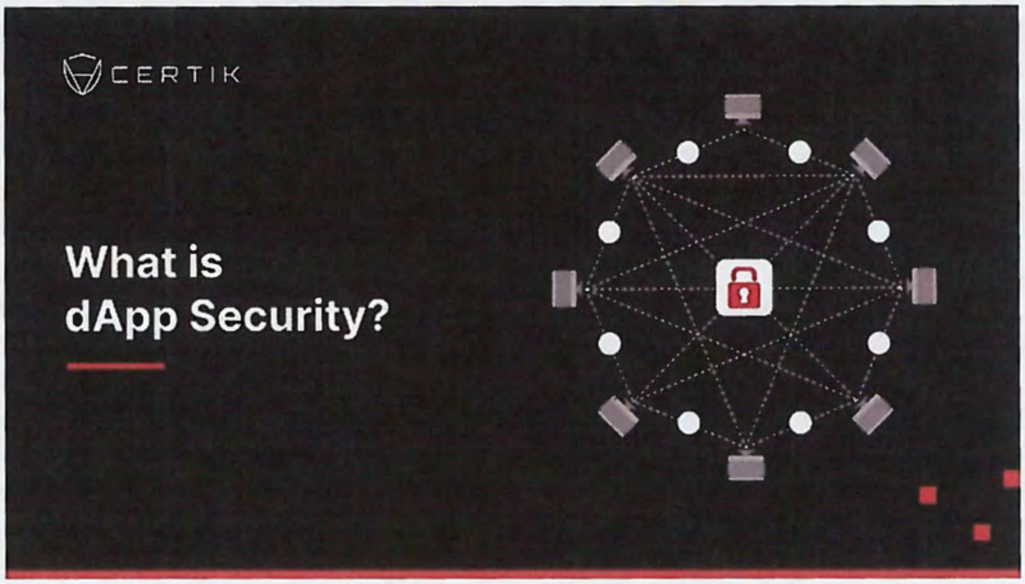


Blogs

What is an Online Attack?

One of the promises of blockchain technology is its ability to bolster the safety of online activity through the greater security afforded by decentralization. However, despite this added security, it is naive to assume that online attacks will just go away. With that in mind, this post provides an overview of some of the most infamous online attacks, how they intersect with blockchain technology, and some of the ways of defending against such attacks.

5/8/2022

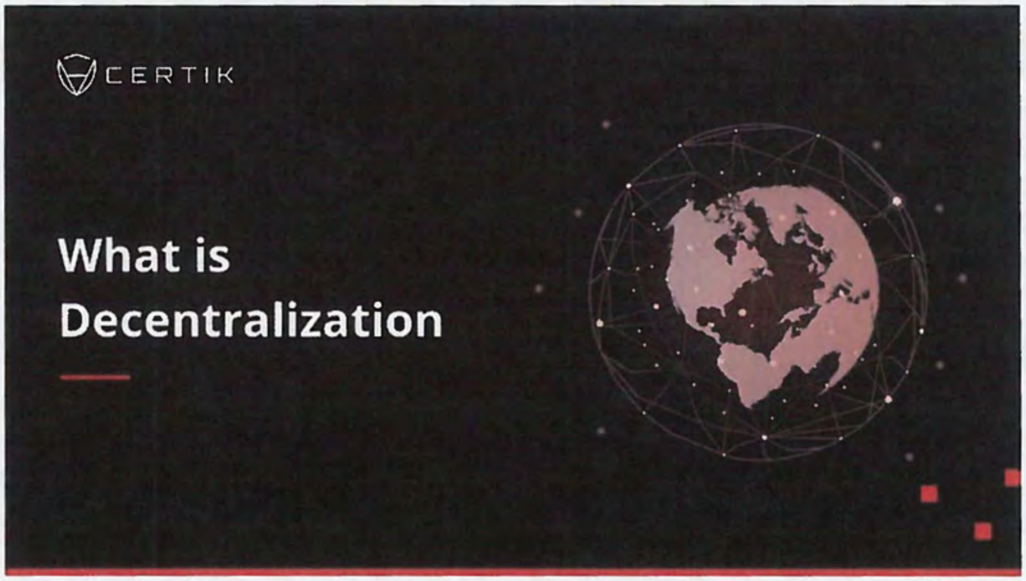


Blogs

What is dApp Security?

The systematic set up of decentralized apps unfortunately leaves them susceptible to hackers in some situations. As more businesses migrate to dApps and other cloud-based structures, it is important to keep safety and security in mind. Even as technology changes, cybercriminals will look for ways to infiltrate it.

4/8/2022



Blogs

What is Decentralization?

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF FLORIDA
PENSACOLA DIVISION

COIN CENTER; PATRICK O'SULLIVAN; JOHN DOE; and DAVID HOFFMAN,

Plaintiffs,

v.

JANET YELLEN, in her official capacity as Secretary of the Treasury; DEPARTMENT OF THE TREASURY; ANDREA M. GACKI, in her official capacity as Director of the Office of Foreign Assets Control; and OFFICE OF FOREIGN ASSETS CONTROL,

Defendants.

Case No.
3:22-cv-20375-TKW-ZCB

JOINT APPENDIX VOLUME II

EXHIBIT 120

Introduction to Tornado Cash

Tornado Cash is a **fully decentralized non-custodial protocol** allowing private transactions in the crypto-space.

As a decentralized protocol, Tornado.Cash smart contracts have been implemented within the Ethereum blockchain, making them immutable. They can neither be changed nor tampered with. Therefore, nobody - including the original developers - can modify or shut them down. All governance and mining smart contracts are deployed by the community in a decentralized manner.

As a non-custodial protocol, users keep custody of their cryptocurrencies while operating Tornado.Cash. This means that at each deposit, they are provided with the private key enabling the access to the deposited funds, which gives users complete control over their assets.

How is privacy achieved?

Tornado Cash improves transaction privacy by breaking the on-chain link between source and destination addresses. It uses a smart contract that accepts ETH & other tokens deposits from one address and enables their withdrawal from a different address.

To maximize privacy, several steps are recommended, such as the use of a relayer for gas payments to withdraw funds from an address with no pre-existing balance.

More details are available in *Behind the scenes:* How does Tornado.Cash work? & Tips to remain anonymous.

Where does Tornado.cash operate?

Since its inception in 2019, Tornado Cash has been operating **on the Ethereum blockchain**. The protocol has been offering diversified fixed amount pools for six tokens (ETH, DAI, cDAI, USDC, USDT & WBTC) handled by the Ethereum blockchain.

Since June 2021, in addition to the Ethereum blockchain, Tornado Cash smart contracts **have also been deployed on other side-chains & blockchains**. These deployments enabled the tool to either support new tokens or benefit from Layer-2 advantages, such as faster and cheaper transactions.

As of today, Tornado Cash is operating on:

- **Ethereum Blockchain** : **ETH** (Ethereum), **DAI** (Dai), **cDAI** (Compound Dai), **USDC** (USD Coin), **USDT** (Tether) & **WBTC** (Wrapped Bitcoin),
- **Binance Smart Chain**: **BNB** (Binance Coin),
- **Polygon Network**: **MATIC** (Polygon),
- **Gnosis Chain (former xDAI Chain)**: **xDAI** (xDai),
- **Avalanche Mainnet**: **AVAX** (Avalanche),
- **Optimism**, as a Layer-2 for **ETH** (Ethereum),
- **Arbitrum One**, as a Layer-2 **ETH** (Ethereum).

Until December 2021, the protocol included an anonymity mining system for some of these tokens, allowing its users to earn a governance token (**TORN**). Users were able to ultimately earn TORN on the Blockchain network by depositing in the ETH, DAI, cDAI & WBTC pools.

More information on Anonymity mining & Tornado.Cash token is *available.*

Thanks to the TORN token, Tornado Cash users can actively participate in shaping the protocol. The community has a strong weight regarding the evolution of Tornado Cash and the improvement of its features. Indeed, protocol parameters & token distribution are completely under the community's control through this governance.

All pools mentioned above can be accessed on tornadocash.eth.link. They operate **under the principle of fixed-amount deposits & withdrawals**. It means that each token has 2 to 4 different pools, allowing transactions of only 2 to 4 different fixed amounts *(e.g. ETH has four different pools, one for each of these amounts: 0.1, 1, 10 & 100 ETH)*.

Tornado Cash Nova

With the **release of Tornado Cash Nova** (beta version) on December 2021, an **upgraded pool with unique new features** has been added to the protocol. Users are no longer constrained by fixed-amount transactions. With the addition of Tornado Cash Nova, they can benefit from the use of **an arbitrary amount pool & shielded transfers**.

Tornado Cash Nova operates on the Gnosis Chain (former xDai Chain) as a Layer2 to optimize speed and cost. It allows **deposits and withdrawals of completely customized amounts in ETH**. This pool also enables shielded transactions where users can **transfer the custody of their token while remaining in the pool**.

Tornado Cash Nova (beta version) can be accessed on nova.tornadocash.eth.link. You can find further informations related to the functioning of Tornado Cash Nova in the dedicated section of our docs.

How does Tornado.Cash run?

Codes behind Tornado.Cash functioning - smart contacts, circuits & toolchain - are fully **open-**

sourced. Working as a DAO (Decentralized Autonomous Organization), Tornado.Cash governance and mining smart contracts are deployed by its community.

The protocol also functions with zk-SNARK, which enables zero-knowledge proofs allowing users to demonstrate possession of information without needing to reveal it. The use of this technology is based **on open-source research made by Zcash team with the help of the Ethereum community**. To set up zk-SNARK initial keys, Tornado.Cash Trusted Setup Community was launched in May 2020 & accounts for 1114 contributions. This significant number of contributors makes it impossible to compromise the protocol by faking zero-knowledge proofs.

User interface is hosted on **IPFS** (InterPlanetary File System) by the community, minimizing risks of data deletion. Indeed, the interface will work as long as at least one user is hosting it.

EXHIBIT 130

NISTIR 8202

Blockchain Technology Overview

Dylan Yaga
Peter Mell
Nik Roby
Karen Scarfone

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8202

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8202

Blockchain Technology Overview

Dylan Yaga
Peter Mell
Computer Security Division
Information Technology Laboratory

Nik Roby
G2, Inc.
Annapolis Junction, MD

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8202

October 2018

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Internal Report 8202
66 pages (October 2018)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8202

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: nistir8202-comments@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company, or government). At their basic level, they enable a community of users to record transactions in a shared ledger within that community, such that under normal operation of the blockchain network no transaction can be changed once published. This document provides a high-level technical overview of blockchain technology. The purpose is to help readers understand how blockchain technology works.

Keywords

blockchain; consensus model; cryptocurrency; cryptographic hash function; asymmetric-key cryptography; distributed ledger; distributed consensus algorithm; proof of work; proof of stake; round robin; proof of authority; proof of identity; proof of elapsed time; soft fork, hard fork; smart contracts; data oracle.

Acknowledgments

The authors wish to thank all contributors to this publication, and their colleagues who reviewed drafts of this report and contributed technical and editorial additions. This includes NIST staff James Dray, Sandy Ressler, Rick Kuhn, Lee Badger, Eric Trapnell, Mark Trapnell, James Shook and Michael Davidson.

Additional thanks to all the people and organizations who submitted comments during the public comment period.

Audience

This publication is designed for readers with little or no knowledge of blockchain technology who wish to understand at a high level how it works. It is not intended to be a technical guide; the discussion of the technology provides a conceptual understanding. Note that some examples, figures, and tables are simplified to fit the audience.

Trademark Information

All registered trademarks and trademarks belong to their respective organizations.

Executive Summary

Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company, or government). At their basic level, they enable a community of users to record transactions in a shared ledger within that community, such that under normal operation of the blockchain network no transaction can be changed once published. In 2008, the blockchain idea was combined with several other technologies and computing concepts to create modern cryptocurrencies: electronic cash protected through cryptographic mechanisms instead of a central repository or authority. The first such blockchain based cryptocurrency was Bitcoin.

Within the Bitcoin blockchain, information representing electronic cash is attached to a digital address. Bitcoin users can digitally sign and transfer rights to that information to another user and the Bitcoin blockchain records this transfer publicly, allowing all participants of the network to independently verify the validity of the transactions. The Bitcoin blockchain is stored, maintained, and collaboratively managed by a distributed group of participants. This, along with certain cryptographic mechanisms, makes the blockchain resilient to attempts to alter the ledger later (modifying blocks or forging transactions).

Because there are countless news articles and videos describing the "magic" of blockchain technology, this paper aims to describe the method behind the magic (i.e., how blockchain technology works). Arthur C. Clarke once wrote, "Any sufficiently advanced technology is indistinguishable from magic" [1]. Clarke's statement is a perfect representation for the emerging applications of blockchain technology. There is hype around the use of blockchain technology, yet the technology is not well understood. It is not magical; it will not solve all problems. As with all new technology, there is a tendency to want to apply it to every sector in every way imaginable. To help promote correct application, this document provides information necessary to develop a high-level understanding of the technology.

Blockchain technology is the foundation of modern cryptocurrencies, so named because of the heavy usage of cryptographic functions. Users utilize public and private keys to digitally sign and securely transact within the system. For cryptocurrency based blockchain networks which utilize mining (see section 4.1), users may solve puzzles using cryptographic hash functions in hopes of being rewarded with a fixed amount of the cryptocurrency. However, blockchain technology may be more broadly applicable than cryptocurrencies. In this work, we focus on the cryptocurrency use case, since that is the primary use of the technology today; however, there is a growing interest in other sectors.

Organizations considering implementing blockchain technology need to understand fundamental aspects of the technology. For example, what happens when an organization implements a blockchain network and then decides they need to make modifications to the data stored? When using a database, modifying the actual data can be accomplished through a database query and update. Organizations must understand that while changes to the actual blockchain data may be difficult, applications using the blockchain as a data layer work around this by treating later blocks and transactions as updates or modifications to earlier blocks and transactions. This software abstraction allows for modifications to working data, while providing a full history of

changes. Another critical aspect of blockchain technology is how the participants agree that a transaction is valid. This is called "reaching consensus", and there are many models for doing so, each with positives and negatives for particular business cases. It is important to understand that a blockchain is just one part of a solution.

Blockchain implementations are often designed with a specific purpose or function. Example functions include cryptocurrencies, smart contracts (software deployed on the blockchain and executed by computers running that blockchain), and distributed ledger systems between businesses. There has been a constant stream of developments in the field of blockchain technology, with new platforms being announced constantly – the landscape is continuously changing.

There are two general high-level categories for blockchain approaches that have been identified: permissionless, and permissioned. In a permissionless blockchain network anyone can read and write to the blockchain without authorization. Permissioned blockchain networks limit participation to specific people or organizations and allow finer-grained controls. Knowing the differences between these two categories allows an organization to understand which subset of blockchain technologies may be applicable to its needs.

Despite the many variations of blockchain networks and the rapid development of new blockchain related technologies, most blockchain networks use common core concepts. Blockchains are a distributed ledger comprised of blocks. Each block is comprised of a block header containing metadata about the block, and block data containing a set of transactions and other related data. Every block header (except for the very first block of the blockchain) contains a cryptographic link to the previous block's header. Each transaction involves one or more blockchain network users and a recording of what happened, and it is digitally signed by the user who submitted the transaction.

Blockchain technology takes existing, proven concepts and merges them together into a single solution. This document explores the fundamentals of how these technologies work and the differences between blockchain approaches. This includes how the participants in the network come to agree on whether a transaction is valid and what happens when changes need to be made to an existing blockchain deployment. Additionally, this document explores when to consider using a blockchain network.

The use of blockchain technology is not a silver bullet, and there are issues that must be considered such as how to deal with malicious users, how controls are applied, and the limitations of the implementations. Beyond the technology issues that need to be considered, there are operational and governance issues that affect the behavior of the network. For example, in permissioned blockchain networks, described later in this document, there are design issues surrounding what entity or entities will operate and govern the network for the intended user base.

Blockchain technology is still new and should be investigated with the mindset of "how could blockchain technology potentially benefit us?" rather than "how can we make our problem fit into the blockchain technology paradigm?". Organizations should treat blockchain technology like they would any other technological solution at their disposal and use it in appropriate situations.

Table of Contents

List of Appendices

List of Tables and Figures

1 Introduction

Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company or government). At their basic level, they enable a community of users to record transactions in a shared ledger within that community, such that under normal operation of the blockchain network no transaction can be changed once published. In 2008, the blockchain idea was combined with several other technologies and computing concepts to create modern cryptocurrencies: electronic cash protected through cryptographic mechanisms instead of a central repository or authority.

This technology became widely known in 2009 with the launch of the Bitcoin network, the first of many modern cryptocurrencies. In Bitcoin, and similar systems, the transfer of digital information that represents electronic cash takes place in a distributed system. Bitcoin users can digitally sign and transfer their rights to that information to another user and the Bitcoin blockchain records this transfer publicly, allowing all participants of the network to independently verify the validity of the transactions. The Bitcoin blockchain is independently maintained and managed by a distributed group of participants. This, along with cryptographic mechanisms, makes the blockchain resilient to attempts to alter the ledger later (modifying blocks or forging transactions). Blockchain technology has enabled the development of many cryptocurrency systems such as Bitcoin and Ethereum[1]. Because of this, blockchain technology is often viewed as bound to Bitcoin or possibly cryptocurrency solutions in general. However, the technology is available for a broader variety of applications and is being investigated for a variety of sectors.

The numerous components of blockchain technology along with its reliance on cryptographic primitives and distributed systems can make it challenging to understand. However, each component can be described simply and used as a building block to understand the larger complex system. Blockchains can be informally defined as:

> Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance). New blocks are replicated across copies of the ledger within the network, and any conflicts are resolved automatically using established rules.

[1] Bitcoin and Ethereum are mentioned here since they are listed as the top two cryptocurrencies on market capitalization websites

1.1 Background and History

The core ideas behind blockchain technology emerged in the late 1980s and early 1990s. In 1989, Leslie Lamport developed the Paxos protocol, and in 1990 submitted the paper *The Part-Time Parliament* [2] to ACM Transactions on Computer Systems; the paper was finally published in a 1998 issue. The paper describes a consensus model for reaching agreement on a result in a network of computers where the computers or network itself may be unreliable. In 1991, a signed chain of information was used as an electronic ledger for digitally signing documents in a way that could easily show none of the signed documents in the collection had been changed [3]. These concepts were combined and applied to electronic cash in 2008 and described in the paper, *Bitcoin: A Peer to Peer Electronic Cash System* [4], which was published pseudonymously by Satoshi Nakamoto, and then later in 2009 with the establishment of the Bitcoin cryptocurrency blockchain network. Nakamoto's paper contained the blueprint that most modern cryptocurrency schemes follow (although with variations and modifications). Bitcoin was just the first of many blockchain applications.

Many electronic cash schemes existed prior to Bitcoin (e.g., ecash and NetCash), but none of them achieved widespread use. The use of a blockchain enabled Bitcoin to be implemented in a distributed fashion such that no single user controlled the electronic cash and no single point of failure existed; this promoted its use. Its primary benefit was to enable direct transactions between users without the need for a trusted third party. It also enabled the issuance of new cryptocurrency in a defined manner to those users who manage to publish new blocks and maintain copies of the ledger; such users are called *miners* in Bitcoin. The automated payment of the miners enabled distributed administration of the system without the need to organize. By using a blockchain and consensus-based maintenance, a self-policing mechanism was created that ensured that only valid transactions and blocks were added to the blockchain.

In Bitcoin, the blockchain enabled users to be pseudonymous. This means that users are anonymous, but their account identifiers are not; additionally, all transactions are publicly visible. This has effectively enabled Bitcoin to offer pseudo-anonymity because accounts can be created without any identification or authorization process (such processes are typically required by Know-Your-Customer (KYC) laws).

Since Bitcoin was pseudonymous, it was essential to have mechanisms to create trust in an environment where users could not be easily identified. Prior to the use of blockchain technology, this trust was typically delivered through intermediaries trusted by both parties. Without trusted intermediaries, the needed trust within a blockchain network is enabled by four key characteristics of blockchain technology, described below:

- **Ledger** – the technology uses an append only ledger to provide full transactional history. Unlike traditional databases, transactions and values in a blockchain are not overridden.
- **Secure** – blockchains are cryptographically secure, ensuring that the data contained within the ledger has not been tampered with, and that the data within the ledger is attestable.
- **Shared** – the ledger is shared amongst multiple participants. This provides transparency across the node participants in the blockchain network.

- **Distributed** – the blockchain can be distributed. This allows for scaling the number of nodes of a blockchain network to make it more resilient to attacks by bad actors. By increasing the number of nodes, the ability for a bad actor to impact the consensus protocol used by the blockchain is reduced.

For blockchain networks that allow anyone to anonymously create accounts and participate (called permissionless blockchain networks), these capabilities deliver a level of trust amongst parties with no prior knowledge of one another; this trust can enable individuals and organizations to transact directly, which may result in transactions being delivered faster and at lower costs. For a blockchain network that more tightly controls access (called permissioned blockchain networks), where some trust may be present among users, these capabilities help to bolster that trust.

1.2 Purpose and Scope

This document provides a high-level technical overview of blockchain technology. It looks at different categories of implementation approaches. It discusses the components of blockchain technology and provides diagrams and examples when possible. It discusses, at a high-level, some consensus models used in blockchain networks. It also provides an overview of how blockchain technology changes (known as forking) affect the blockchain network. It provides details on how blockchain technology was extended beyond attestable transactions to include attestable application processes known as smart contracts. It also touches on some of the limitations and misconceptions surrounding the technology. Finally, this document presents several areas that organizations should consider when investigating blockchain technology. It is intended to help readers to understand the technologies which comprise blockchain networks.

1.3 Notes on Terms

The terminology for blockchain technology varies from one implementation to the next – to talk about the technology, generic terms will be used. Throughout this document the following terms will be used:

- *Blockchain* – the actual ledger
- *Blockchain technology* – a term to describe the technology in the most generic form
- *Blockchain network* – the network in which a blockchain is being used
- *Blockchain implementation* – a specific blockchain
- *Blockchain network user* – a person, organization, entity, business, government, etc. which is utilizing the blockchain network
- *Node* – an individual system within a blockchain network
    - *Full node* – a node that stores the entire blockchain, ensures transactions are valid
        - *Publishing node* – a full node that also publishes new blocks
    - *Lightweight node* – a node that does not store or maintain a copy of the blockchain and must pass their transactions to full nodes

1.4 Results of the Public Comment Period

This document has seen substantial revision in response to the public comments received. Part of the revising process was to tighten the scope, and to provide a more foundational document as an introduction to the technology. Please note that several sections present in the draft (7.1.2 - Permissioned Use Cases, 7.2.2 - Permissionless Use Cases, and 8 - Blockchain Platforms) are not present in the published version. These topics were made explicitly out of scope for this document because the rapidly changing landscape and areas of interest around this technology, as well as the ever-increasing number of platforms, would make these sections out of place in such a foundational document. The topics in these sections are still being considered for future works.

Additionally, section 8.1.2 – Bitcoin Cash contained an erroneous and unverified statement which was not identified and removed during initial editing of the draft. Since this section has been removed, this issue is now addressed.

1.5 Document Structure

The rest of this document is organized as follows:

- **Section 2** discusses the high-level categorization of blockchain technology: permissionless and permissioned.
- **Section 3** defines the high-level components of a blockchain network architecture, including hashes, transactions, ledgers, blocks, and blockchains.
- **Section 4** discusses several consensus models employed by blockchain technology.
- **Section 5** introduces the concept of forking.
- **Section 6** discusses smart contracts.
- **Section 7** discusses several limitations as well as misconceptions surrounding blockchain technology.
- **Section 8** discusses various application considerations, as well as provides additional considerations from government, academia, and technology enthusiasts.
- **Section 9** is the conclusion.
- **Appendix A** provides a list of acronyms and abbreviations used in the document.
- **Appendix B** contains a glossary for selected terms defined in the document.
- **Appendix C** lists the references used throughout the document.

2 Blockchain Categorization

Blockchain networks can be categorized based on their permission model, which determines who can maintain them (e.g., publish blocks). If anyone can publish a new block, it is *permissionless*. If only particular users can publish blocks, it is *permissioned*. In simple terms, a permissioned blockchain network is like a corporate intranet that is controlled, while a permissionless blockchain network is like the public internet, where anyone can participate. Permissioned blockchain networks are often deployed for a group of organizations and individuals, typically referred to as a consortium. This distinction is necessary to understand as it impacts some of the blockchain components discussed later in this document.

2.1 Permissionless

Permissionless blockchain networks are decentralized ledger platforms open to anyone publishing blocks, without needing permission from any authority. Permissionless blockchain platforms are often open source software, freely available to anyone who wishes to download them. Since anyone has the right to publish blocks, this results in the property that anyone can read the blockchain as well as issue transactions on the blockchain (through including those transactions within published blocks). Any blockchain network user within a permissionless blockchain network can read and write to the ledger. Since permissionless blockchain networks are open to all to participate, malicious users may attempt to publish blocks in a way that subverts the system (discussed in detail later). To prevent this, permissionless blockchain networks often utilize a multiparty agreement or 'consensus' system (see Section 4) that requires users to expend or maintain resources when attempting to publish blocks. This prevents malicious users from easily subverting the system. Examples of such consensus models include proof of work (see Section 4.1) and proof of stake (see Section 4.2) methods. The consensus systems in permissionless blockchain networks usually promote non-malicious behavior through rewarding the publishers of protocol-conforming blocks with a native cryptocurrency.

2.2 Permissioned

Permissioned blockchain networks are ones where users publishing blocks must be authorized by some authority (be it centralized or decentralized). Since only authorized users are maintaining the blockchain, it is possible to restrict read access and to restrict who can issue transactions. Permissioned blockchain networks may thus allow anyone to read the blockchain or they may restrict read access to authorized individuals. They also may allow anyone to submit transactions to be included in the blockchain or, again, they may restrict this access only to authorized individuals. Permissioned blockchain networks may be instantiated and maintained using open source or closed source software.

Permissioned blockchain networks can have the same traceability of digital assets as they pass through the blockchain, as well as the same distributed, resilient, and redundant data storage system as a permissionless blockchain networks. They also use consensus models for publishing blocks, but these methods often do not require the expense or maintenance of resources (as is the case with current permissionless blockchain networks). This is because the establishment of one's identity is required to participate as a member of the permissioned blockchain network; those maintaining the blockchain have a level of trust with each other, since they were all authorized to publish blocks and since their authorization can be revoked if they misbehave. Consensus models in permissioned blockchain networks are then usually faster and less computationally expensive.

Permissioned blockchain networks may also be used by organizations that need to more tightly control and protect their blockchain. However, if a single entity controls who can publish blocks, the users of the blockchain will need to have trust in that entity. Permissioned blockchain networks may also be used by organizations that wish to work together but may not fully trust one another. They can establish a permissioned blockchain network and invite business partners to record their transactions on a shared distributed ledger. These organizations can determine the consensus model to be used, based on how much they trust one another. Beyond trust, permissioned blockchain networks provide transparency and insight that may help better inform business decisions and hold misbehaving parties accountable. This can explicitly include auditing and oversight entities making audits a constant occurrence versus a periodic event.

Some permissioned blockchain networks support the ability to selectively reveal transaction information based on a blockchain network users identity or credentials. With this feature, some degree of privacy in transactions may be obtained. For example, it could be that the blockchain records that a transaction between two blockchain network users took place, but the actual contents of transactions is only accessible to the involved parties.

Some permissioned blockchain networks require all users to be authorized to send and receive transactions (they are not anonymous, or even pseudo-anonymous). In such systems parties work together to achieve a shared business process with natural disincentives to commit fraud or otherwise behave as a bad actor (since they can be identified). If bad behavior were to occur, it is well known where the organizations are incorporated, what legal remedies are available and how to pursue those remedies in the relevant judicial system.

3 Blockchain Components

Blockchain technology can seem complex; however, it can be simplified by examining each component individually. At a high level, blockchain technology utilizes well-known computer science mechanisms and cryptographic primitives (cryptographic hash functions, digital signatures, asymmetric-key cryptography) mixed with record keeping concepts (such as append only ledgers). This section discusses each individual main component: cryptographic hash functions, transactions, asymmetric-key cryptography, addresses, ledgers, blocks, and how blocks are chained together.

3.1 Cryptographic Hash Functions

An important component of blockchain technology is the use of cryptographic hash functions for many operations. *Hashing* is a method of applying a cryptographic hash function to data, which calculates a relatively unique output (called a *message digest*, or just *digest*) for an input of nearly any size (e.g., a file, text, or image). It allows individuals to independently take input data, hash that data, and derive the same result – proving that there was no change in the data. Even the smallest change to the input (e.g., changing a single bit) will result in a completely different output digest. Table 1 shows simple examples of this.

Cryptographic hash functions have these important security properties:

1. They are *preimage resistant.* This means that they are one-way; it is computationally infeasible to compute the correct input value given some output value (e.g., given a digest, find x such that hash(x) = digest).
2. They are *second preimage resistant.* This means one cannot find an input that hashes to a specific output. More specifically, cryptographic hash functions are designed so that given a specific input, it is computationally infeasible to find a second input which produces the same output (e.g., given x, find y such that hash(x) = hash(y)). The only approach available is to exhaustively search the input space, but this is computationally infeasible to do with any chance of success.
3. They are *collision resistant.* This means that one cannot find two inputs that hash to the same output. More specifically, it is computationally infeasible to find any two inputs that produce the same digest (e.g., find an x and y which hash(x) = hash(y)).

A specific cryptographic hash function used in many blockchain implementations is the Secure Hash Algorithm (SHA) with an output size of 256 bits (SHA-256). Many computers support this algorithm in hardware, making it fast to compute. SHA-256 has an output of 32 bytes (1 byte = 8 bits, 32 bytes = 256 bits), generally displayed as a 64-character hexadecimal string (see Table 1 below).

This means that there are $2^{256} \approx 10^{77}$, or 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936 possible digest values. The algorithm for SHA-256, as well as others, is specified in Federal Information Processing Standard (FIPS) 180-4 [5]. The NIST Secure Hashing website [6] contains FIPS specifications for all NIST-approved hashing algorithms.

Table 1: Examples of Input Text and Corresponding SHA-256 Digest Values

Input Text	SHA-256 Digest Value
1	0x6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdffd6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

Since there are an infinite number of possible input values and a finite number of possible output digest values, it is possible but highly unlikely to have a collision where hash(*x*) = hash(*y*) (i.e., the hash of two different inputs produces the same digest). SHA-256 is said to be collision resistant, since to find a collision in SHA-256, one would have to execute the algorithm, on average, about 2^{128} times (which is 340 undecillions, or more precisely 340,282,366,920,938,463,463,374,607,431,768,211,456; roughly 3.402 x 10^{38}).

To put this into perspective, the hash rate (hashes per second) of the entire Bitcoin network in 2015 was 300 quadrillion hashes per second (300,000,000,000,000,000/s) [7]. At that rate, it would take the entire Bitcoin network roughly 35,942,991,748,521 (roughly 3.6 x 10^{13}) years[2] to manufacture a collision (note that the universe is estimated to be 1.37 x 10^{10} years old)[3]. Even if any such input *x* and *y* that produce the same digest, it would be also very unlikely for both inputs to be valid in the context of the blockchain network (i.e., *x* and *y* are both valid transactions).

Within a blockchain network, cryptographic hash functions are used for many tasks, such as:

- Address derivation – discussed in section 3.4.
- Creating unique identifiers.
- Securing the block data – a publishing node will hash the block data, creating a digest that will be stored within the block header.
- Securing the block header – a publishing node will hash the block header. If the blockchain network utilizes a proof of work consensus model (see Section 4.1), the publishing node will need to hash the block header with different nonce values (see Section 3.1.1) until the puzzle requirements have been fulfilled. The current block header's hash digest will be included within the next block's header, where it will secure the current block header data.

Because the block header includes a hash representation of the block data, the block data itself is

[2] Calculation: 2^{128}/((((300000000000000000×60) (hash per second -> minute)
×60) (minute -> hour)
×24) (hour -> day)
×365.25) (day -> year) = 35942991748521.060268986932617580573454677584269188193 years
https://www.wolframalpha.com/input/?i=2%5E128%2F(300000000000000000+*+60+*+60+*+24+*+365.25)

[3] As estimated by measurements made by the Wilkinson Microwave Anisotropy Probe
https://map.gsfc.nasa.gov/universe/uni_age.html

also secured when the block header digest is stored in the next block.

There are many families of cryptographic hash functions utilized in blockchain technology (SHA-256 is not the only one), such as Keccak (which was selected by NIST as the winner of a competition to create the SHA-3 hashing standard), as well as RIPEMD-160.[8]

3.1.1 Cryptographic Nonce

A cryptographic nonce is an arbitrary number that is only used once. A cryptographic nonce can be combined with data to produce different hash digests per nonce:

$$\text{hash (data + nonce) = digest}$$

Only changing the nonce value provides a mechanism for obtaining different digest values while keeping the same data. This technique is utilized in the proof of work consensus model (see Section 4.1).

3.2 Transactions

A *transaction* represents an interaction between parties. With cryptocurrencies, for example, a transaction represents a transfer of the cryptocurrency between blockchain network users. For business-to-business scenarios, a transaction could be a way of recording activities occurring on digital or physical assets. Figure 1 shows a notional example of a cryptocurrency transaction. Each block in a blockchain can contain zero or more transactions. For some blockchain implementations, a constant supply of new blocks (even with zero transactions) is critical to maintain the security of the blockchain network; by having a constant supply of new blocks being published, it prevents malicious users from ever "catching up" and manufacturing a longer, altered blockchain (see Section 4.7).

The data which comprises a transaction can be different for every blockchain implementation, however the mechanism for transacting is largely the same. A blockchain network user sends information to the blockchain network. The information sent may include the sender's address (or another relevant identifier), sender's public key, a digital signature, transaction inputs and transaction outputs.

A single cryptocurrency transaction typically requires at least the following information, but can contain more:

- **Inputs** – The inputs are usually a list of the digital assets to be transferred. A transaction will reference the source of the digital asset (providing provenance) – either the previous transaction where it was given to the sender, or for the case of new digital assets, the origin event. Since the input to the transaction is a reference to past events, the digital assets do not change. In the case of cryptocurrencies this means that value cannot be added or removed from existing digital assets. Instead, a single digital asset can be split into multiple new digital assets (each with lesser value) or multiple digital assets can be combined to form fewer new digital assets (with a correspondingly greater value). The splitting or joining of assets will be specified within the transaction output.

The sender must also provide proof that they have access to the referenced inputs, generally by digitally signing the transaction – proving access to the private key.

- **Outputs** – The outputs are usually the accounts that will be the recipients of the digital assets along with how much digital asset they will receive. Each output specifies the number of digital assets to be transferred to the new owner(s), the identifier of the new owner(s), and a set of conditions the new owners must meet to spend that value. If the digital assets provided are more than required, the extra funds must be explicitly sent back to the sender (this is a mechanism to "make change").

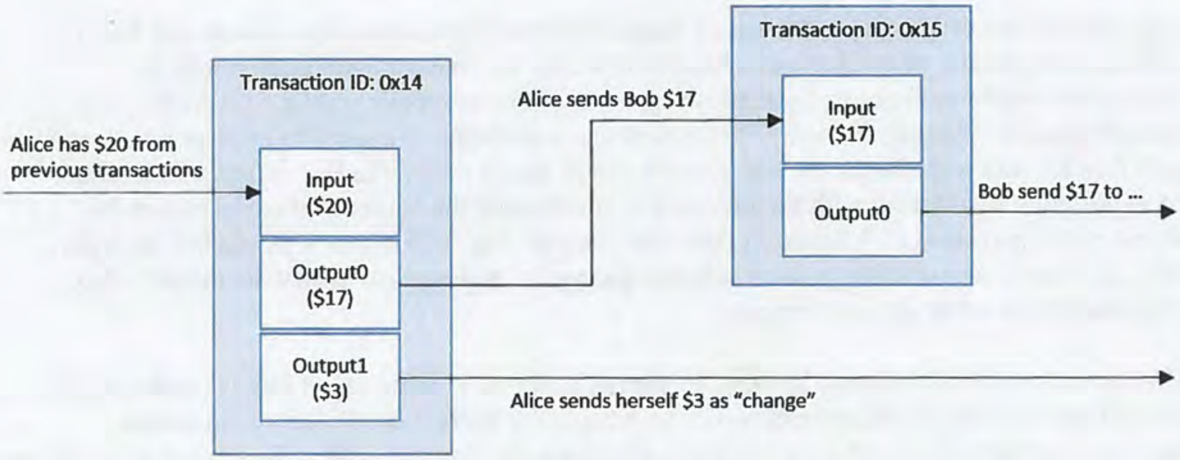


Figure 1 - Example Cryptocurrency Transaction

While primarily used to transfer digital assets, transactions can be more generally used to transfer data. In a simple case, someone may simply want to permanently and publicly post data on the blockchain. In the case of smart contract systems, transactions can be used to send data, process that data, and store some result on the blockchain. For example, a transaction can be used to change an attribute of a digitized asset such as the location of a shipment within a blockchain technology-based supply chain system.

Regardless of how the data is formed and transacted, determining the validity and authenticity of a transaction is important. The validity of a transaction ensures that the transaction meets the protocol requirements and any formalized data formats or smart contract requirements specific to the blockchain implementation. The authenticity of a transaction is also important, as it determines that the sender of digital assets had access to those digital assets. Transactions are typically digitally signed by the sender's associated private key (asymmetric-key cryptography is briefly discussed in Section 3.3) and can be verified at any time using the associated public key.

3.3 Asymmetric-Key Cryptography

Blockchain technology uses asymmetric-key cryptography[4] (also referred to as public key cryptography). Asymmetric-key cryptography uses a pair of keys: a public key and a private key that are mathematically related to each other. The public key is made public without reducing the security of the process, but the private key must remain secret if the data is to retain its cryptographic protection. Even though there is a relationship between the two keys, the private key cannot efficiently be determined based on knowledge of the public key. One can encrypt with a private key and then decrypt with the public key. Alternately, one can encrypt with a public key and then decrypt with a private key.

Asymmetric-key cryptography enables a trust relationship between users who do not know or trust one another, by providing a mechanism to verify the integrity and authenticity of transactions while at the same time allowing transactions to remain public. To do this, the transactions are 'digitally signed'. This means that a private key is used to encrypt a transaction such that anyone with the public key can decrypt it. Since the public key is freely available, encrypting the transaction with the private key proves that the signer of the transaction has access to the private key. Alternately, one can encrypt data with a user's public key such that only users with access to the private key can decrypt it. A drawback is that asymmetric-key cryptography is often slow to compute.

This contrasts with symmetric-key cryptography in which a single secret key is used to both encrypt and decrypt. With symmetric-key cryptography users must already have a trust relationship established with one another to exchange the pre-shared key. In a symmetric system, any encrypted data that can be decrypted with the pre-shared key confirms it was sent by another user with access to the pre-shared key; no user without access to the pre-shared key will be able to view the decrypted data. Compared to asymmetric-key cryptography, symmetric-key cryptography is very fast to compute. Because of this, when one claims to be encrypting something using asymmetric-key cryptography, oftentimes the data is encrypted with symmetric-key cryptography and then the symmetric-key is encrypted using asymmetric-key cryptography. This 'trick' can greatly speed up asymmetric-key cryptography.

Here is a summary of the use of asymmetric-key cryptography in many blockchain networks:

- Private keys are used to digitally sign transactions.
- Public keys are used to derive addresses.
- Public keys are used to verify signatures generated with private keys.
- Asymmetric-key cryptography provides the ability to verify that the user transferring value to another user is in possession of the private key capable of signing the transaction.

[4] FIPS Publication 186-4, Digital Signature Standard [9] specifies a common algorithm for digital signing used in blockchain technologies: Elliptic Curve Digital Signature Algorithm (ECDSA).

Some permissioned blockchain networks can leverage a business's existing public key infrastructure for asymmetric-key cryptography to provide user credentials – rather than having each blockchain network user manage their own asymmetric-keys. This is done by utilizing existing directory services and using that information within the blockchain network. Blockchain networks which utilize an existing directory service can access it via existing protocols, such as the Lightweight Directory Access Protocol (LDAP) [10], and utilize the information from the directory natively, or import it into an internal certificate authority within the blockchain network.

3.4 Addresses and Address Derivation

Some blockchain networks make use of an *address*, which is a short, alphanumeric string of characters derived from the blockchain network user's public key using a cryptographic hash function, along with some additional data (e.g., version number, checksums). Most blockchain implementations make use of addresses as the "to" and "from" endpoints in a transaction. Addresses are shorter than the public keys and are not secret. One method to generate an address is to create a public key, applying a cryptographic hash function to it, and converting the hash to text:

public key → cryptographic hash function → address

Each blockchain implementation may implement a different method to derive an address. For permissionless blockchain networks, which allow anonymous account creation, a blockchain network user can generate as many asymmetric-key pairs, and therefore addresses as desired, allowing for a varying degree of pseudo-anonymity. Addresses may act as the public-facing identifier in a blockchain network for a user, and oftentimes an address will be converted into a QR code (Quick Response Code, a 2-dimensional bar code which can contain arbitrary data) for easier use with mobile devices.

Figure 2 - A QR code example which has encoded the text "NISTIR 8202 - Blockchain Technology Overview QR code example"

Blockchain network users may not be the only source of addresses within blockchain networks. It is necessary to provide a method of accessing a smart contract once it has been deployed within a blockchain network. For Ethereum, smart contracts are accessible via a special address called a contract account. This account address is created when a smart contract is deployed (the address for a contract account is deterministically computed from the smart contract creator's address [11]). This contract account allows for the contract to be executed whenever it receives a transaction, as well as create additional smart contracts in turn.

3.4.1 Private Key Storage

With some blockchain networks (especially with permissionless blockchain networks), users must manage and securely store their own private keys. Instead of recording them manually, they often use software to securely store them. This software is often referred to as a *wallet*. The wallet can store private keys, public keys, and associated addresses. It may also perform other functions, such as calculating the total number of digital assets a user may have.

If a user loses a private key, then any digital asset associated with that key is lost, because it is computationally infeasible to regenerate the same private key. If a private key is stolen, the attacker will have full access to all digital assets controlled by that private key. The security of private keys is so important that many users use special secure hardware to store them; alternatively, users may take advantage of an emerging industry of private key escrow services. These key escrow services can also satisfy KYC laws in addition to storing private keys as users must provide proof of their identity when creating an account.

Private key storage is an extremely important aspect of blockchain technology. When it is reported in the news that "Cryptocurrency XYZ was stolen from…", it almost certainly means some private keys were found and used to sign a transaction sending the money to a new account, not that the blockchain network itself was compromised. Note that because blockchain data cannot generally be changed, once a criminal steals a private key and publicly transfers the associated funds to another account, that transaction generally cannot be undone.

3.5 Ledgers

A *ledger* is a collection of transactions. Throughout history, pen and paper ledgers have been used to keep track of the exchange of goods and services. In modern times, ledgers have been stored digitally, often in large databases owned and operated by a centralized trusted third party (i.e., the owner of the ledger) on behalf of a community of users. These ledgers with centralized ownership can be implemented in a centralized or distributed fashion (i.e., just one server or a coordinating cluster of servers).

There is growing interest in exploring having distributed ownership of the ledger. Blockchain technology enables such an approach using both distributed ownership as well as a distributed physical architecture. The distributed physical architecture of blockchain networks often involve a much larger set of computers than is typical for centrally managed distributed physical architecture. The growing interest in distributed ownership of ledgers is due to possible trust, security, and reliability concerns related to ledgers with centralized ownership:

- Centrally owned ledgers may be lost or destroyed; a user must trust that the owner is properly backing up the system.
  - A blockchain network is distributed by design, creating many backup copies all updating and syncing to the same ledger data between peers. A key benefit to blockchain technology is that every user can maintain their own copy of the ledger. Whenever new full nodes join the blockchain network, they reach out to discover other full nodes and request a full copy of the blockchain network's ledger, making loss or destruction of the ledger difficult.
    Note – certain blockchain implementations provide the capability to support concepts such as private transactions or private channels. Private transactions facilitate the delivery of information only to those nodes participating in a transaction and not the entire network.
- Centrally owned ledgers may be on a homogeneous network, where all software, hardware and network infrastructure may be the same. Because of this characteristic, the overall system resiliency may be reduced since an attack on one part of the network will work on everywhere.
  - A blockchain network is a heterogeneous network, where the software, hardware and network infrastructure are all different. Because of the many differences between nodes on the blockchain network, an attack on one node is not guaranteed to work on other nodes.
- Centrally owned ledgers may be located entirely in specific geographic locations (e.g., all in one country). If network outages were to occur in that location, the ledger and services which depend on it may not be available.
  - A blockchain network can be comprised of geographically diverse nodes which may be found around the world. Because of this, and the blockchain network working in a peer-to-peer fashion, it is resilient to the loss of any node, or even an entire region of nodes.
- The transactions on a centrally owned ledger are not made transparently and may not be valid; a user must trust that the owner is validating each received transaction.
  - A blockchain network must check that all transactions are valid; if a malicious node was transmitting invalid transactions, others would detect and ignore them, preventing the invalid transactions from propagating throughout the blockchain network.
- The transaction list on a centrally owned ledger may not be complete; a user must trust that the owner is including all valid transactions that have been received.
  - A blockchain network holds all accepted transactions within its distributed ledger. To build a new block, a reference must be made to a previous block – therefore building on top of it. If a publishing node did not include a reference to the latest block, other nodes would reject it.
- The transaction data on a centrally owned ledger may have been altered; a user must trust that the owner is not altering past transactions.

  - A blockchain network utilizes cryptographic mechanisms such as digital signatures and cryptographic hash functions to provide tamper evident and tamper resistant ledgers.

- The centrally owned system may be insecure; a user must trust that the associated computer systems and networks are receiving critical security patches and have implemented best practices for security. The system may be breached and have had personal information stolen because of insecurities.
  - A blockchain network, due to the distributed nature, provides no centralized point of attack. Generally, information on a blockchain network is publicly viewable, and offers nothing to steal. To attack blockchain network users, an attacker would need to individually target them. Targeting the blockchain itself would be met with the resistance of the honest nodes present in the system. If an individual node was not patched, it would only affect that node – not the system overall.

3.6 Blocks

Blockchain network users submit candidate transactions to the blockchain network via software (desktop applications, smartphone applications, digital wallets, web services, etc.). The software sends these transactions to a node or nodes within the blockchain network. The chosen nodes may be non-publishing full nodes as well as publishing nodes. The submitted transactions are then propagated to the other nodes in the network, but this by itself does not place the transaction in the blockchain. For many blockchain implementations, once a pending transaction has been distributed to nodes, it must then wait in a queue until it is added to the blockchain by a publishing node.

Transactions are added to the blockchain when a publishing node publishes a block. A *block* contains a block header and block data. The block header contains metadata for this block. The block data contains a list of validated and authentic transactions which have been submitted to the blockchain network. Validity and authenticity is ensured by checking that the transaction is correctly formatted and that the providers of digital assets in each transaction (listed in the transaction's 'input' values) have each cryptographically signed the transaction. This verifies that the providers of digital assets for a transaction had access to the private key which could sign over the available digital assets. The other full nodes will check the validity and authenticity of all transactions in a published block and will not accept a block if it contains invalid transactions.

It should be noted that every blockchain implementation can define its own data fields; however, many blockchain implementations utilize data fields like the following:

- Block Header
  - The block number, also known as block height in some blockchain networks.
  - The previous block header's hash value.
  - A hash representation of the block data (different methods can be used to accomplish this, such as a generating a Merkle tree (defined in Appendix B), and storing the root hash, or by utilizing a hash of all the combined block data).
  - A timestamp.

- The size of the block.
- The nonce value. For blockchain networks which utilize mining, this is a number which is manipulated by the publishing node to solve the hash puzzle (see Section 4.1 for details). Other blockchain networks may or may not include it or use it for another purpose other than solving a hash puzzle.

- Block Data
  - A list of transactions and ledger events included within the block.
  - Other data may be present.

3.7 Chaining Blocks

Blocks are chained together through each block containing the hash digest of the previous block's header, thus forming the *blockchain*. If a previously published block were changed, it would have a different hash. This in turn would cause all subsequent blocks to also have different hashes since they include the hash of the previous block. This makes it possible to easily detect and reject altered blocks. Figure 3 shows a generic chain of blocks.

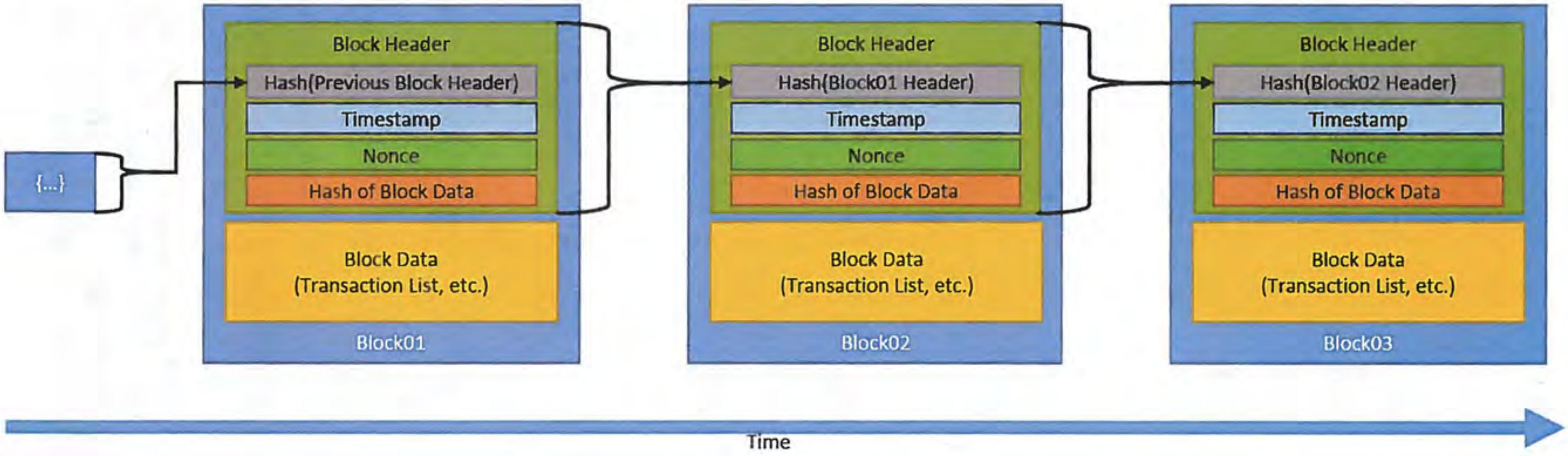


Figure 3: Generic Chain of Blocks

4 Consensus Models

A key aspect of blockchain technology is determining which user publishes the next block. This is solved through implementing one of many possible consensus models. For permissionless blockchain networks there are generally many publishing nodes competing at the same time to publish the next block. They usually do this to win cryptocurrency and/or transaction fees. They are generally mutually distrusting users that may only know each other by their public addresses. Each publishing node is likely motivated by a desire for financial gain, not the well-being of the other publishing nodes or even the network itself.

In such a situation, why would a user propagate a block that another user is attempting to publish? Also, who resolves conflicts when multiple nodes publish a block at approximately the same time? To make this work, blockchain technologies use *consensus models* to enable a group of mutually distrusting users to work together.

When a user joins a blockchain network, they agree to the initial state of the system. This is recorded in the only pre-configured block, the *genesis block*. Every blockchain network has a published genesis block and every block must be added to the blockchain after it, based on the agreed-upon consensus model. Regardless of the model, however, each block must be valid and thus can be validated independently by each blockchain network user. By combining the initial state and the ability to verify every block since then, users can independently agree on the current state of the blockchain. Note that if there were ever two valid chains presented to a full node, the default mechanism in most blockchain networks is that the 'longer' chain is viewed as the correct one and will be adopted; this is because it has had the most amount of work put into it. This happens frequently with some consensus models and will be discussed in detail.

The following properties are then in place:

- The initial state of the system is agreed upon (e.g., the genesis block).
- Users agree to the consensus model by which blocks are added to the system.
- Every block is linked to the previous block by including the previous block header's hash digest (except for the first 'genesis' block, which has no previous block and for which the hash of the previous block header is usually set to all zeros).
- Users can verify every block independently.

In practice, software handles everything and the users do not need to be aware of these details.

A key feature of blockchain technology is that there is no need to have a trusted third party provide the state of the system—every user within the system can verify the system's integrity. To add a new block to the blockchain, all nodes must come to a common agreement over time; however, some temporary disagreement is permitted. For permissionless blockchain networks, the consensus model must work even in the presence of possibly malicious users since these users might attempt to disrupt or take over the blockchain. Note that for permissioned blockchain networks legal remedies may be used if a user acts maliciously.

In some blockchain networks, such as permissioned, there may exist some level of trust between publishing nodes. In this case, there may not be the need for a resource intensive (computation time, investment, etc.) consensus model to determine which participant adds the next block to the chain. Generally, as the level of trust increases, the need for resource usage as a measure of generating trust decreases. For some permissioned blockchain implementations, the view of consensus extends beyond ensuring validity and authenticity of the blocks but encompasses the entire systems of checks and validations from the proposal of a transaction, to its final inclusion on a block.

In the following sections, several consensus models as well as the most common conflict resolution approach are discussed.

4.1 Proof of Work Consensus Model

In the proof of work (PoW) model, a user publishes the next block by being the first to solve a computationally intensive puzzle. The solution to this puzzle is the "proof" they have performed work. The puzzle is designed such that solving the puzzle is difficult but checking that a solution is valid is easy. This enables all other full nodes to easily validate any proposed next blocks, and any proposed block that did not satisfy the puzzle would be rejected.

A common puzzle method is to require that the hash digest of a block header be less than a target value. Publishing nodes make many small changes to their block header (e.g., changing the nonce) trying to find a hash digest that meets the requirement. For each attempt, the publishing node must compute the hash for the entire block header. Hashing the block header many times becomes a computationally intensive process. The target value may be modified over time to adjust the difficulty (up or down) to influence how often blocks are being published.

For example, Bitcoin, which uses the proof of work model, adjusts the puzzle difficulty every 2016 blocks to influence the block publication rate to be around once every ten minutes. The adjustment is made to the difficulty level of the puzzle, and essentially either increases or decreases the number of leading zeros required. By increasing the number of leading zeros, it increases the difficulty of the puzzle, because any solution must be less than the difficulty level – meaning there are fewer possible solutions. By decreasing the number of leading zeros, it decreases the difficulty level, because there are more possible solutions. This adjustment is to maintain the computational difficulty of the puzzle, and therefore maintain the core security mechanism of the Bitcoin network. Available computing power increases over time, as does the number of publishing nodes, so the puzzle difficulty is generally increasing.

Adjustments to the difficulty target aim to ensure that no entity can take over block production, but as a result the puzzle solving computations require significant resource consumption. Due to the significant resource consumption of some proof of work blockchain networks, there is a move to add publishing nodes to areas where there is a surplus supply of cheap electricity.

An important aspect of this model is that the work put into a puzzle does not influence one's likelihood of solving the current or future puzzles because the puzzles are independent. This means that when a user receives a completed and valid block from another user, they are incentivized to discard their current work and to start building off the newly received block instead because they know the other publishing nodes will be building off it.

As an example, consider a puzzle where, using the SHA-256 algorithm, a computer must find a hash value meeting the following target criteria (known as the difficulty level):

```
SHA256("blockchain" + Nonce) = Hash Digest starting with "000000"
```

In this example, the text string "`blockchain`" is appended with a nonce value and then the hash digest is calculated. The nonce values used will be numeric values only. This is a relatively easy puzzle to solve and some sample output follows:

```
SHA256("blockchain0") =
0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab938
(not solved)

SHA256("blockchain1") =
0xdb0b9c1cb5e9c680dfff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10
(not solved)

…

SHA256("blockchain10730895") =
0x000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d67587
(solved)
```

To solve this puzzle, it took 10,730,896 guesses (completed in 54 seconds on relatively old hardware, starting at 0 and testing one value at a time).

In this example, each additional "leading zero" value increases the difficulty. By increasing the target by one additional leading zero ("0000000"), the same hardware took 934,224,175 guesses to solve the puzzle (completed in 1 hour, 18 minutes, 12 seconds):

```
SHA256("blockchain934224174") =
0x0000000e2ae7e4240df80692b7e586ea7a977eacbd031819d0e603257edb3a81
```

There is currently no known shortcut to this process; publishing nodes must expend computation effort, time, and resources to find the correct nonce value for the target. Often the publishing nodes attempt to solve this computationally difficult puzzle to claim a reward of some sort (usually in the form of a cryptocurrency offered by the blockchain network). The prospect of being rewarded for extending and maintaining the blockchain is referred to as a reward system or incentive model.

Once a publishing node has performed this work, they send their block with a valid nonce to full nodes in the blockchain network. The recipient full nodes verify that the new block fulfills the puzzle requirement, then add the block to their copy of the blockchain and resend the block to their peer nodes. In this manner, the new block gets quickly distributed throughout the network of participating nodes. Verification of the nonce is easy since only a single hash needs to be done to check to see if it solves the puzzle.

For many proof of work based blockchain networks, publishing nodes tend to organize themselves into "pools" or "collectives" whereby they work together to solve puzzles and split the reward. This is possible because work can be distributed between two or more nodes across a collective to share the workload and rewards. Splitting the example program into quarters, each node can take an equal amount of the nonce value range to test:

- `Node 1: check nonce 0000000000 to 0536870911`
- `Node 2: check nonce 0536870912 to 1073741823`
- `Node 3: check nonce 1073741824 to 1610612735`
- `Node 4: check nonce 1610612736 to 2147483647`

The following result was the first to be found to solve the puzzle:

```
SHA256("blockchain1700876653") =
0x00000003ba55d20c9cbd1b6fb34dd81c3553360ed918d07acf16dc9e75d7c7f1
```

This is a completely new nonce, but still one that solved the puzzle. It took 90,263,918 guesses (completed in 10 minutes, 14 seconds). Dividing up the work amongst many more machines yields much better results, as well as more consistent rewards in a proof of work model.

The use of a computationally difficult puzzle helps to combat the "Sybil Attack" – a computer security attack (not limited to blockchain networks) where an attacker can create many nodes (i.e., creating multiple identities) to gain influence and exert control. The proof of work model combats this by having the focus of network influence being the amount of computational power (hardware, which costs money) mixed with a lottery system (the most hardware increases likelihood but does not guarantee it) versus in network identities (which are generally costless to create).

4.2 Proof of Stake Consensus Model

The proof of stake (PoS) model is based on the idea that the more stake a user has invested into the system, the more likely they will want the system to succeed, and the less likely they will want to subvert it. Stake is often an amount of cryptocurrency that the blockchain network user has invested into the system (through various means, such as by locking it via a special transaction type, or by sending it to a specific address, or holding it within special wallet software). Once staked, the cryptocurrency is generally no longer able to be spent. Proof of stake blockchain networks use the amount of stake a user has as a determining factor for publishing new blocks. Thus, the likelihood of a blockchain network user publishing a new block is tied to the ratio of their stake to the overall blockchain network amount of staked cryptocurrency.

With this consensus model, there is no need to perform resource intensive computations (involving time, electricity, and processing power) as found in proof of work. Since this consensus model utilizes fewer resources, some blockchain networks have decided to forego a block creation reward; these systems are designed so that all the cryptocurrency is already distributed among users rather than new cryptocurrency being generated at a constant pace. In such systems, the reward for block publication is then usually the earning of user provided transaction fees.

The methods for how the blockchain network uses the stake can vary. Here we discuss four approaches: random selection of staked users, multi-round voting, coin aging systems and delegate systems. Regardless of the exact approach, users with more stake are more likely to publish new blocks.

When the choice of block publisher is a random choice (sometimes referred to as *chain-based proof of stake*), the blockchain network will look at all users with stake and choose amongst them based on their ratio of stake to the overall amount of cryptocurrency staked. So, if a user had 42 % of the entire blockchain network stake they would be chosen 42 % of the time; those with 1 % would be chosen 1 % of the time.

When the choice of block publisher is a multi-round voting system (sometime referred to as *Byzantine fault tolerance proof of stake* [12]) there is added complexity. The blockchain network will select several staked users to create proposed blocks. Then all staked users will cast a vote for a proposed block. Several rounds of voting may occur before a new block is decided upon. This method allows all staked users to have a voice in the block selection process for every new block.

When the choice of block publisher is through a coin age system referred to as a *coin age proof of stake,* staked cryptocurrency has an *age* property. After a certain amount of time (such as 30 days) the staked cryptocurrency can *count* towards the owning user being selected to publish the next block. The staked cryptocurrency then has its *age* reset, and it cannot be used again until after the requisite time has passed. This method allows for users with more stake to publish more blocks, but to not dominate the system – since they have a cooldown timer attached to every cryptocurrency coin *counted* towards creating blocks. Older coins and larger groups of coins will increase the probability of being chosen to publish the next block. To prevent stakeholders from hoarding aged cryptocurrencies, there is generally a built-in maximum to the probability of winning.

When the choice of block publisher is through a delegate system, users vote for nodes to become publishing nodes – therefore creating blocks on their behalf. Blockchain network users' voting power is tied to their stake so the larger the stake, the more weight the vote has. Nodes who receive the most votes become publishing nodes and can validate and publish blocks. Blockchain network users can also vote against an established publishing node, to try to remove them from the set of publishing nodes. Voting for publishing nodes is continuous and remaining a publishing node can be quite competitive. The threat of losing publishing node status, and therefore rewards and reputation is constant so publishing nodes are incentivized to not act maliciously. Additionally, blockchain network users vote for delegates, who participate in the governance of the blockchain. Delegates will propose changes, and improvements, which will be voted on by blockchain network users.

It is worth noting that a problem known as "nothing at stake" may arise from some proof of stake algorithms. If multiple competing blockchains were to exist at some point (because of a temporary ledger conflict as discussed in Section 4.7), a staked user could act on every such competing chain – since it is essentially free to do so. The staked user may do this as a way of increasing their odds of earning a reward. This can cause multiple blockchain branches to continue to grow without being reconciled into a singular branch for extended periods of time.

Under proof of stake systems, the "rich" can more easily stake more of the digital assets, earning themselves more digital assets; however, to obtain the majority of digital assets within a system to "control" it is generally cost prohibitive.

4.3 Round Robin Consensus Model

Round Robin is a consensus model that is used by some permissioned blockchain networks. Within this model of consensus, nodes take turns in creating blocks. Round Robin Consensus has a long history grounded in distributed system architecture. To handle situations where a publishing node is not available to publish a block on its turn, these systems may include a time limit to enable available nodes to publish blocks so that unavailable nodes will not cause a halt in block publication. This model ensures no one node creates the majority of the blocks. It benefits from a straightforward approach, lacks cryptographic puzzles, and has low power requirements.

Since there is a need for trust amongst nodes, round robin does not work well in the permissionless blockchain networks used by most cryptocurrencies. This is because malicious nodes could continuously add additional nodes to increase their odds of publishing new blocks. In the worst case, they could use this to subvert the correct operation of the blockchain network.

4.4 Proof of Authority/Proof of Identity Consensus Model

The proof of authority (also referred to as proof of identity) consensus model relies on the partial trust of publishing nodes through their known link to real world identities. Publishing nodes must have their identities proven and verifiable within the blockchain network (e.g., identifying documents which have been verified and notarized and included on the blockchain). The idea is that the publishing node is staking its identity/reputation to publish new blocks. Blockchain network users directly affect a publishing node's reputation based on the publishing node's behavior. Publishing nodes can lose reputation by acting in a way that the blockchain network users disagree with, just as they can gain reputation by acting in a manner that the blockchain network users agree with. The lower the reputation, the less likelihood of being able to publish a block. Therefore, it is in the interest of a publishing node to maintain a high reputation. This algorithm only applies to permissioned blockchain networks with high levels of trust.

4.5 Proof of Elapsed Time Consensus Model

Within the proof of elapsed time (PoET) consensus model, each publishing node requests a wait time from a secure hardware time source within their computer system. The secure hardware time source will generate a random wait time and return it to the publishing node software. Publishing nodes take the random time they are given and become idle for that duration. Once a publishing node wakes up from the idle state, it creates and publishes a block to the blockchain network, alerting the other nodes of the new block; any publishing node that is still idle will stop waiting, and the entire process starts over.

This model requires ensuring that a random time was used, since if the time to wait was not selected at random a malicious publishing node would just wait the minimum amount of time by default to dominate the system. This model also requires ensuring that the publishing node waited the actual time and did not start early. These requirements are being solved by executing software in a trusted execution environment found on some computer processors (such as Intel's Software Guard Extensions[5], or AMD's Platform Security Processor[6], or ARM's TrustZone[7]).

Verified and trusted software can run in these secure execution environments and cannot be altered by outside programs. A publishing node would query software running in this secure environment for a random time and then wait for that time to pass. After waiting the assigned time, the publishing node could request a signed certificate that the publishing node waited the randomly assigned time. The publishing node then publishes the certificate along with the block.

[5] Intel SGX - https://software.intel.com/en-us/sgx

[6] AMD Secure Technology - https://www.amd.com/en/technologies/security

[7] ARM TrustZone - https://www.arm.com/products/silicon-ip-security

4.6 Consensus Comparison Matrix

Name	Goals	Advantages	Disadvantages	Domains	Implementations
Proof of work (PoW)	To provide a barrier to publishing blocks in the form of a computationally difficult puzzle to solve to enable transactions between untrusted participants.	Difficult to perform denial of service by flooding network with bad blocks. Open to anyone with hardware to solve the puzzle.	Computationally intensive (by design), power consumption, hardware arms race. Potential for 51 % attack by obtaining enough computational power.	Permissionless cryptocurrencies	Bitcoin, Ethereum, many more
Proof of stake (PoS)	To enable a less computationally intensive barrier to publishing blocks, but still enable transactions between untrusted participants.	Less computationally intensive than PoW. Open to anyone who wishes to stake cryptocurrencies. Stakeholders control the system.	Stakeholders control the system. Nothing to prevent formation of a pool of stakeholders to create a centralized power. Potential for 51 % attack by obtaining enough financial power.	Permissionless cryptocurrencies	Ethereum Casper, Krypton
Delegated PoS	To enable a more efficient consensus model through a 'liquid democracy' where participants vote (using cryptographically signed messages) to elect and revoke the rights of delegates to validate and secure the blockchain.	Elected delegates are economically incentivized to remain honest More computationally efficient than PoW	Less node diversity than PoW or pure PoS consensus implementations Greater security risk for node compromise due to constrained set of operating nodes As all delegates are 'known' there may an incentive for block producers to collude and accept bribes, compromising the security of the system	Permissionless cryptocurrencies Permissioned Systems	Bitshares, Steem, Cardano, EOS

Name	Goals	Advantages	Disadvantages	Domains	Implementations
Round Robin	Provide a system for publishing blocks amongst approved/trusted publishing nodes	Low computational power. Straightforward to understand.	Requires large amount of trust amongst publishing nodes.	Permissioned Systems	MultiChain
Proof of Authority/Identity	To create a centralized consensus process to minimize block creation and confirmation rate	Fast confirmation time Allows for dynamic block production rates Can be used in sidechains to blockchain networks which utilize another consensus model	Relies on the assumption that the current validating node has not been compromised Leads to centralized points of failure The reputation of a given node is subject to potential for high tail-risk as it could be compromised at any time.	Permissioned Systems, Hybrid (sidechain) Systems	Ethereum Kovan testnet, POA Chain, various permissioned systems using Parity
Proof of Elapsed Time (PoET)	To enable a more economic consensus model for blockchain networks, at the expense of deeper security guarantees associated with PoW.	Less computationally expensive than PoW	Hardware requirement to obtain time. Assumes the hardware clock used to derive time is not compromised Given speed-of-late latency limits, true time synchronicity is essentially impossible in distributed systems [13]	Permissioned Networks	Hyperledger Sawtooth

4.7 Ledger Conflicts and Resolutions

As discussed previously, for some blockchain networks it is possible that multiple blocks will be published at approximately the same time. This can cause differing versions of a blockchain to exist at any given moment; these must be resolved quickly to have consistency in the blockchain network. In this section, we discuss how these situations are generally handled.

With any distributed network, some systems within the network will be behind on information or have alternative information. This depends on network latency between nodes and the proximity of groups of nodes. Permissionless blockchain networks are more prone to have conflicts due to their openness and number of competing publishing nodes. A major part of agreeing on the state of the blockchain network (coming to consensus) is resolving conflicting data.

For example:

- `node_A` creates `block_n(A)` with transactions #1, 2 and 3. `node_A` distributes it to some nodes.
- `node_B` creates `block_n(B)` with transactions #1, 2 and 4. `node_B` distributes it to some nodes.
- **There is a conflict.**
    - o `block_n` will not be the same across the network.
        - ▪ `block_n(A)` contains transaction #3, but not transaction #4.
        - ▪ `block_n(B)` contains transaction #4, but not transaction #3.

Conflicts temporarily generate different versions of the blockchain, which is depicted in Figure 4. These differing versions are not "wrong"; rather, they were created with the information each node had available. The competing blocks will likely contain different transactions, so those with `block_n(A)` may see transfers of digital assets not present in `block_n(B)`. If the blockchain network deals with cryptocurrency, then a situation may occur where some cryptocurrency may both be spent and unspent, depending on which version of the blockchain is being viewed.

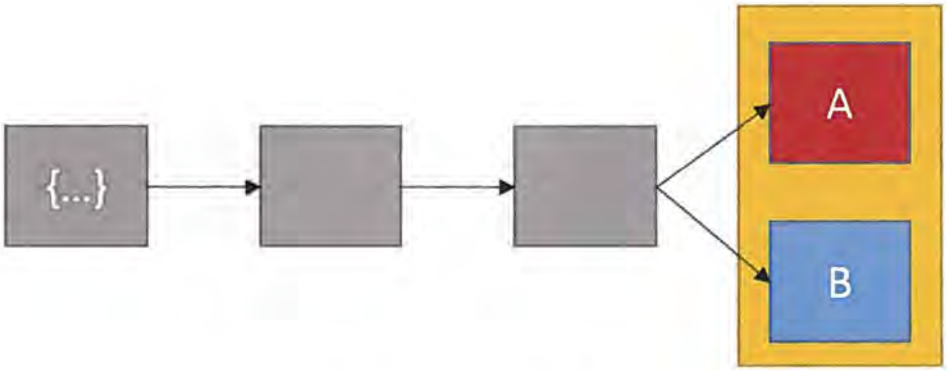


Figure 4: Ledger in Conflict

Conflicts are usually quickly resolved. Most blockchain networks will wait until the next block is published and use that chain as the "official" blockchain, thus adopting the "longer blockchain". As in Figure 5, the blockchain containing `block_n(B)` becomes the "official" chain, as it got the next valid block. Any transaction that was present in `block_n(A)`, the orphaned block, but not present in the `block_n(B)` chain, is returned to the pending transaction pool (which is where all transactions which have not been included within a block reside). Note that this set of pending transactions is maintained locally at each node as there is no central server in the architecture.

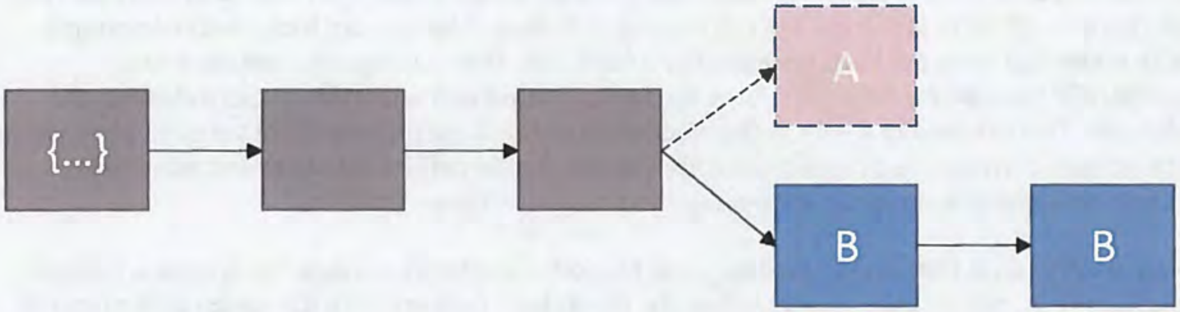


Figure 5: The chain with block_n(B) adds the next block, the chain with block_n(A) is now orphaned

Due to the possibility of blocks being overwritten, a transaction is not usually accepted as confirmed until several additional blocks have been created on top of the block containing the relevant transaction. The acceptance of a block is often probabilistic rather than deterministic since blocks can be superseded. The more blocks that have been built on top of a published block, the more likely it is that the initial block will not be overwritten.

Hypothetically, a node in a proof of work blockchain network with enormous amounts of computing power could start at the genesis block and create a longer chain than the currently existing chain, thereby wiping out the entire blockchain history. This does not happen in practice due to the prohibitively large amount of resources that this would require. Also, some blockchain implementations lock specific older blocks within the blockchain software by creating checkpoints to ensure that this can never happen.

5 Forking

Performing changes and updating technology can be difficult at the best of times. For permissionless blockchain networks which are comprised of many users, distributed around the world, and governed by the consensus of the users, it becomes extremely difficult. Changes to a blockchain network's protocol and data structures are called *forks*. They can be divided into two categories: *soft forks* and *hard forks*. For a soft fork, these changes are backwards compatible with nodes that have not been updated. For a hard fork, these changes are not backwards compatible because the nodes that have not been updated will reject the blocks following the changes. This can lead to a split in the blockchain network creating multiple versions of the same blockchain. Permissioned blockchain networks, due to the publishing nodes and users being known, can mitigate the issues of forking by requiring software updates.

Note that the term fork is also used by some blockchain networks to describe temporary ledger conflicts (e.g., two or more blocks within the blockchain network with the same block number) as described in Section 4.7. While this is a fork in the ledger, it is temporary and does not stem from a software change.

5.1 Soft Forks

A *soft fork* is a change to a blockchain implementation that is backwards compatible. Non-updated nodes can continue to transact with updated nodes. If no (or very few) nodes upgrade, then the updated rules will not be followed.

An example of a soft fork occurred on Bitcoin when a new rule was added to support escrow[8] and time-locked refunds. In 2014, a proposal was made to repurpose an operation code that performed no operation (OP_NOP2) to CHECKLOCKTIMEVERIFY, which allows a transaction output to be made spendable at a point in the future [14]. For nodes that implement this change, the node software will perform this new operation, but for nodes that do not support the change, the transaction is still valid, and execution will continue as if a NOP [9] had been executed.

A fictional example of a soft fork would be if a blockchain decided to reduce the size of blocks (for example from 1.0 MB to 0.5 MB). Updated nodes would adjust the block size and continue to transact as normal; non-updated nodes would see these blocks as valid – since the change made does not violate their rules (i.e., the block size is under their maximum allowed). However, if a non-updated node were to create a block with a size greater than 0.5 MB, updated nodes would reject them as invalid.

5.2 Hard Forks

A *hard fork* is a change to a blockchain implementation that is not backwards compatible. At a

[8] Funds placed into a third party to be disseminated based on conditions (via multi-signature transactions)

[9] NOP meaning No Operation

given point in time (usually at a specific block number), all publishing nodes will need to switch to using the updated protocol. Additionally, all nodes will need to upgrade to the new protocol so that they do not reject the newly formatted blocks. Non-updated nodes cannot continue to transact on the updated blockchain because they are programmed to reject any block that does not follow their version of the block specification.

Publishing nodes that do not update will continue to publish blocks using the old format. User nodes that have not updated will reject the newly formatted blocks and only accept blocks with the old format. This results in two versions of the blockchain existing simultaneously. Note that users on different hard fork versions cannot interact with one another. It is important to note that while most hard forks are intentional, software errors may produce unintentional hard forks.

A well-known example of a hard fork is from Ethereum. In 2016, a smart contract was constructed on Ethereum called the Decentralized Autonomous Organization (DAO). Due to flaws in how the smart contract was constructed, an attacker extracted Ether, the cryptocurrency used by Ethereum, resulting in the theft of $50 million [15]. A hard fork proposal was voted on by Ether holders, and the clear majority of users agreed to hard fork and create a new version of the blockchain, without the flaw, and that also returned the stolen funds.

With cryptocurrencies, if there is a hard fork and the blockchain splits then users will have independent currency on both forks (having double the number of coins in total). If all the activity moves to the new chain, the old one may eventually not be used since the two chains are not compatible (they will be independent currency systems). In the case of the Ethereum hard fork, the clear majority of support moved to the new fork, the old fork was renamed Ethereum Classic and continued operating.

5.3 Cryptographic Changes and Forks

If flaws are found in the cryptographic technologies within a blockchain network, the only solution may be to create a hard fork, depending on the significance of the flaw. For example, if a flaw was found in the underlying algorithms, there could be a fork requiring all future clients to use a stronger algorithm. Switching to a new hashing algorithm could pose a significant practical problem because it could invalidate all existing specialized mining hardware.

Hypothetically, if SHA-256 were discovered to have a flaw, blockchain networks that utilize SHA-256 would need a hard fork to migrate to a new hash algorithm. The block that switched over to the new hash algorithm would “lock” all previous blocks into SHA-256 (for verification), and all new blocks would need to utilize the new hashing algorithm. There are many cryptographic hash algorithms, and blockchain networks can make use of whichever suits their needs. For example, while Bitcoin uses SHA-256, Ethereum uses Keccak-256 [8].

One possibility for the need to change cryptographic features present in a blockchain network would be the development of a practical quantum computer system, which would be capable of greatly weakening (and in some cases, rendering useless) existing cryptographic algorithms. NIST Internal Report (NISTIR) 8105, Report on Post-Quantum Cryptography [16] provides a table describing the impact of quantum computing on common cryptographic algorithms. Table 2 replicates this table.

Table 2: Impact of Quantum Computing on Common Cryptographic Algorithms

Cryptographic Algorithm	Type	Purpose	Impact from Large-Scale Quantum Computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	N/A	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

The cryptographic algorithms utilized within most blockchain technologies for asymmetric-key pairs will need to be replaced if a powerful quantum computer becomes a reality. This is because algorithms that rely on the computational complexity of integer factorization (such as RSA) or work on solving discrete logarithms (such as DSA and Diffie-Hellman) are very susceptible to being broken by quantum computing. The hashing algorithms used by blockchain networks are much less susceptible to quantum computing attacks but are still weakened.

6 Smart Contracts

The term smart contract dates to 1994, defined by Nick Szabo as "a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries." [17].

Smart contracts extend and leverage blockchain technology. A *smart contract* is a collection of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the blockchain network (e.g., Ethereum's smart contracts, Hyperledger Fabric's chaincode). The smart contract is executed by nodes within the blockchain network; all nodes that execute the smart contract must derive the same results from the execution, and the results of execution are recorded on the blockchain.

Blockchain network users can create transactions which send data to public functions offered by a smart contract. The smart contract executes the appropriate method with the user provided data to perform a service. The code, being on the blockchain, is also tamper evident and tamper resistant and therefore can be used (among other purposes) as a trusted third party. A smart contract can perform calculations, store information, expose properties to reflect a publicly exposed state and, if appropriate, automatically send funds to other accounts. It does not necessarily even have to perform a financial function. For example, the authors of this document have created an Ethereum smart contract that publicly generate trustworthy random numbers [18]. It is important to note that not every blockchain can run smart contracts.

The smart contract code can represent a multi-party transaction, typically in the context of a business process. In a multi-party scenario, the benefit is that this can provide attestable data and transparency that can foster trust, provide insight that can enable better business decisions, reduce costs from reconciliation that exists in traditional business to business applications, and reduce the time to complete a transaction.

Smart contracts must be deterministic, in that given an input they will always produce the same output based on that input. Additionally, all the nodes executing the smart contract must agree on the new state that is obtained after the execution. To achieve this, smart contracts cannot operate on data outside of what is directly passed into it (e.g., smart contracts cannot obtain web services data from within the smart contract – it would need to be passed in as a parameter). Any smart contract which uses data from outside the context of its own system is said to use an 'Oracle' (the oracle problem is described in section 7.3).

For many blockchain implementations, the publishing nodes execute the smart contract code simultaneously when publishing new blocks. There are some blockchain implementations in which there are publishing nodes which do not execute smart contract code, but instead validate the results of the nodes that do. For smart contract enabled permissionless blockchain networks (such as Ethereum) the user issuing a transaction to a smart contract will have to pay for the cost of the code execution. There is a limit on how much execution time can be consumed by a call to a smart contract, based on the complexity of the code. If this limit is exceeded, execution stops, and the transaction is discarded. This mechanism not only rewards the publishers for executing the smart contract code, but also prevents malicious users from deploying and then accessing smart contracts that will perform a denial of service on the publishing nodes by consuming all resources (e.g., using infinite loops).

For smart contract enabled permissioned blockchain networks, such as those utilizing Hyperledger Fabric's chaincode, there may not be a requirement for users to pay for smart contract code execution. These networks are designed around having known participants, and other methods of preventing bad behavior can be employed (e.g., revoking access).

7 Blockchain Limitations and Misconceptions

There is a tendency to overhype and overuse most nascent technology. Many projects will attempt to incorporate the technology, even if it is unnecessary. This stems from the technology being relatively new and not well understood, the technology being surrounded by misconceptions, and the fear of missing out. Blockchain technology has not been immune. This section highlights some of the limitations and misconceptions of blockchain technology.

7.1 Immutability

Most publications on blockchain technology describe blockchain ledgers as being immutable. However, this is not strictly true. They are tamper evident and tamper resistant which is a reason they are trusted for financial transactions. They cannot be considered completely immutable, because there are situations in which the blockchain can be modified. In this section we will look at different ways in which the concept of immutability for blockchain ledgers can be violated.

The chain of blocks itself cannot be considered completely immutable. For some blockchain implementations, the most recently published, or 'tail' blocks are subject to being replaced (by a longer, alternative chain with different 'tail' blocks). As noted earlier, most blockchain networks use the strategy of adopting the longest chain (the one with the most amount of work put into it) as truth when there are multiple competing chains. If two chains are competing, but each include their own unique sequence of tail blocks, whichever is longer will be adopted. However, this does not mean that the transactions within the replaced blocks are lost – rather they may have been included in a different block or returned to the pending transaction pool. This degree of weak immutability for tail blocks is why most blockchain network users wait several block creations before considering a transaction to be valid.

For permissionless blockchain networks, the adoption of a longer, alternate chain of blocks could be the result of a form of attack known as a 51 % attack [19]. For this, the attacker simply garners enough resources to outpace the block creation rate of rest of the blockchain network (holding more than 51 % of the resources applied towards producing new blocks). Depending on the size of the blockchain network, this could be a very cost prohibitive attack carried out by state level actors [20]. The cost to perform this type of attack increases the further back in the blockchain the attacker wishes to make a change. This attack is not technically difficult (e.g., it is just repeating the normal process of the blockchain implementation, but with selected transactions either included or omitted, and at a faster pace), it is just expensive.

For permissioned blockchain networks, this attack can be mitigated. There is generally an owner or consortium of blockchain network users who allow publishing nodes to join the blockchain network and remove publishing nodes from the blockchain network, which gives them a great amount of control. There is less likely to be competing chains since the owner or consortium can force publishing nodes to collaborate fairly since non-cooperating publishing nodes can simply have their privileges removed. There are likely additional legal contracts in place for the blockchain network users which may include clauses for misconduct and the ability to take legal action. While this control is useful to prevent misconduct, it means that any number of blocks can be replaced through legitimate methods if desired by the owner or consortium.

7.2 Users Involved in Blockchain Governance

The governance of blockchain networks deals with the rules, practices and processes by which the blockchain network is directed and controlled. A common misconception is that blockchain networks are systems without control and ownership. The phrase "no one controls a blockchain!" is often exclaimed. This is not strictly true. Permissioned blockchain networks are generally setup and run by an owner or consortium, which governs the blockchain network. Permissionless blockchain networks are often governed by blockchain network users, publishing nodes, and software developers. Each group has a level of control that affects the direction of the blockchain network's advancement.

Software developers create the blockchain software that is utilized by a blockchain network. Since most blockchain technologies are open source, it is possible to inspect the source code, and compile it independently; it is even possible to create separate but compatible software as a means of bypassing pre-compiled software released by developers. However, not every user will have the ability to do this, which means that the developer of the blockchain software will play a large role in the blockchain network's governance. These developers may act in the interest of the community at large and are held accountable. For example, in 2013 Bitcoin developers released a new version of the most popular Bitcoin client which introduced a flaw and started two competing chains of blocks. The developers had to decide to either keep the new version (which had not yet been adopted by everyone) or revert to the old version [21]. Either choice would result in one chain being discarded—and some blockchain network user's transactions becoming invalid. The developers made a choice, reverted to the old version, and successfully controlled the progress of the Bitcoin blockchain.

This example was an unintentional fork; however, developers can purposely design updates to blockchain software to change the blockchain protocol or format. With enough user adoption, a successful fork can be created. Such forks of blockchain software updates are often discussed at length and coordinated with the involved users. For permissionless blockchain networks, this is usually the publishing nodes. There is often a long discussion and adoption period before an event occurs where all users must switch to the newly updated blockchain software at some chosen block to continue recording transactions on the new "main" fork.

For permissionless blockchain networks, although the developers maintain a large degree of influence, users can reject a change by the developers by refusing to install updated software. Of the blockchain network users, the publishing nodes have significant control since they create and publish new blocks. The user base usually adopts the blocks produced by the publishing nodes but is not required to do so. An interesting side effect of this is that permissionless blockchain networks are essentially ruled by the publishing nodes and may marginalize a segment of users by forcing them to adopt changes they may disagree with to stay with the main fork.

For permissioned blockchain networks, control and governance is driven by members of the associated owner or consortium. The consortium can govern who can join the network, when members are removed from the network, coding guidelines for smart contracts, etc.

In summary, the software developers, publishing nodes, and blockchain network users all play a part in the blockchain network governance.

7.3 Beyond the Digital

Blockchain networks work extremely well with the data within their own digital systems. However, when they need to interact with the real world, there are some issues (often called the Oracle Problem [22]). A blockchain network can be a place to record both human input data as well as sensor input data from the real world, but there may be no method to determine if the input data reflects real world events. A sensor could be malfunctioning and recording data that is inaccurate. Humans could record false information (intentionally or unintentionally). These issues are not specific to blockchain networks, but to digital systems overall. However, for blockchain networks that are pseudonymous, dealing with data misrepresentation outside of the digital network can be especially problematic.

For example, if a cryptocurrency transaction took place to purchase a real-world item there is no way to determine within the blockchain network whether the shipment took place, without relying on outside sensor or human input.

Many projects have attempted to address the 'Oracle problem' and create reliable mechanisms to ingest external data in a way that is both trustworthy and accurate. For example, projects like 'Oraclize' provide mechanisms to take web API data and convert it into blockchain readable byte/opcode. Within the context of decentralized applications, these projects may be considered centralized as they provide single points of failure for attackers to compromise. As a result, projects like 'Mineable Oracle Contract' [23] have recently arisen to enable oracle ingestion in a way that is inspired by blockchain technology and built atop established consensus models and economic incentives.

7.4 Blockchain Death

Traditional centralized systems are created and taken down constantly, and blockchain networks will likely not be different. However, because they are decentralized, there is a chance that when a blockchain network "shuts down" it will never be fully shut down, and that there may always be some lingering blockchain nodes running.

A defunct blockchain would not be suitable for a historical record, since without many publishing nodes, a malicious user could easily overpower the few publishing nodes left and redo and replace any number of blocks.

7.5 Cybersecurity

The use of blockchain technology does not remove inherent cybersecurity risks that require thoughtful and proactive risk management. Many of these inherent risks involve a human element. Therefore, a robust cybersecurity program remains vital to protecting the network and participating organizations from cyber threats, particularly as hackers develop more knowledge about blockchain networks and their vulnerabilities.

Existing cybersecurity standards and guidance remain highly relevant for ensuring the security of systems that interface and/or rely on blockchain networks. Subject to certain adjustments to consider specific attributes of blockchain technology, existing standards and guidance provide a strong foundation for protecting blockchain networks from cyberattacks.

In addition to general principles and controls, there are specific cybersecurity standards with relevance to blockchain technology which already exist and are in wide use by many industries. For instance, the NIST Cybersecurity Framework expressly states that it is "not a one-size-fits-all approach to managing cybersecurity risk" because "organizations will continue to have unique risks—different threats, different vulnerabilities, different risk tolerances—and how they implement the practices in the [Framework] will vary." With that said, even though the Framework was not designed for blockchain technology specifically, its standards are broad enough to cover blockchain technology and to help institutions develop policies and processes that identify and control risks affecting blockchain technology.

7.5.1 Cyber and Network-based Attacks

Blockchain technologies are touted as being extremely secure due to the tamper evident and tamper resistant design – once a transaction is committed to the blockchain, it generally cannot be changed. However, this is only true for transactions which have been included in a published block. Transactions that have not yet been included in a published block within the blockchain are vulnerable to several types of attacks. For blockchain networks which have transactional timestamps, spoofing time or adjusting the clock of a member of an ordering service could have positive or negative effects on a transaction, making time and the communication of time an attack vector. Denial of service attacks can be conducted on the blockchain platform or on the smart contract implemented on the platform.

Blockchain networks and their applications are not immune to malicious actors who can conduct network scanning and reconnaissance to discover and exploit vulnerabilities and launch zero-day attacks. In the rush to deploy blockchain-based services, newly coded applications (like smart contracts) may contain new and known vulnerabilities and deployment weaknesses that will be discovered and then attacked through the network just like how websites or applications are attacked today.

7.6 Malicious Users

While a blockchain network can enforce transaction rules and specifications, it cannot enforce a user code of conduct. This is problematic in permissionless blockchain networks, since users are pseudonymous and there is not a one-to-one mapping between blockchain network user identifiers and users of the system. Permissionless blockchain networks often provide a reward (e.g., a cryptocurrency) to motivate users to act fairly; however, some may choose to act maliciously if that provides greater rewards. The largest problem for malicious users is getting enough power (be it a stake in the system, processing power, etc.) to cause damage. Once a large enough malicious collusion is created, malicious mining actions can include:

- Ignoring transactions from specific users, nodes, or even entire countries.
- Creating an altered, alternative chain in secret, then submitting it once the alternative chain is longer than the real chain. The honest nodes will switch to the chain that has the most "work" done (per the blockchain protocol). This could attack the principle of a blockchain network being tamper evident and tamper resistant [24].
- Refusing to transmit blocks to other nodes, essentially disrupting the distribution of information (this is not an issue if the blockchain network is sufficiently decentralized).

While malicious users can be annoyances and create short-term harm, blockchain networks can perform hard forks to combat them. Whether damages done (money lost) would be reversed would be up to the developers and users of the blockchain network.

In addition to there being malicious users of the network, the administrators of the infrastructure for permissioned blockchain networks may also act maliciously. For example, an infrastructure administrator may be able (depending upon the exact configuration) to take over block production, exclude certain users from performing transactions, rewrite block history, double spend coin, delete resources, or re-route or block network connections.

7.7 No Trust

Another common misinterpretation comes from people hearing that there is no "trusted third party" in a blockchain and assuming blockchain networks are "trustless" environments. While there is no trusted third party certifying transactions in permissionless blockchain networks (in permissioned systems it is less clear, as administrators of those systems act as an administrator of trust by granting users admission and permissions), there is still a great deal of trust needed to work within a blockchain network:

- There is trust in the cryptographic technologies utilized. For example, cryptographic algorithms or implementations can have flaws.
- There is trust in the correct and bug free operation of smart contracts, which might have unintended loopholes and flaws.
- There is trust in the developers of the software to produce software that is as bug-free as possible.
- There is trust that most users of the blockchain are not colluding in secret. If a single group or individual can control more than 50 percent of all block creation power, it is possible to subvert a permissionless blockchain network. However, generally obtaining the necessary computational power is prohibitively expensive.
- For blockchain network users not running a full node, there is trust that nodes are accepting and processing transactions fairly.

7.8 Resource Usage

Blockchain technology has enabled a worldwide network where every transaction is verified and the blockchain is kept in sync amongst a multitude of users. For blockchain networks utilizing proof of work, there are many publishing nodes expending large amounts of processing time and, more importantly, consuming a lot of electricity. A proof of work method is an effective solution for "hard to solve, easy to verify" proofs; however, it generally requires significant resource usage. Because of their different applications, and trust models, many permissioned blockchain technologies do not use a resource intensive proof, but rather they utilize different mechanisms to achieve consensus.

The proof of work consensus model is designed for the case where there is little to no trust amongst users of the system. It ensures that publishing nodes cannot game the system[10] by

[10] Use the rules and procedures meant to protect the system to manipulate the system for a desired result.

always being able to solve the puzzles and thereby control the blockchain and the transactions added to it. However, a major concern surrounding the proof of work consensus model is its use of energy in solving the puzzles.

The amount of energy used is often not trivial; for example, some estimate that currently the Bitcoin blockchain network uses around the same amount of electricity as the entire country of Ireland [25]. It has also been speculated that the Bitcoin blockchain network will consume as much electricity as the entire country of Denmark by 2020 [26][27][28]. Software and hardware will continue to improve, resulting in more efficient puzzle solving (reducing the amount of electricity utilized) [29]. However, blockchain networks are also still growing, resulting in harder puzzle difficulty.

An additional strain on resources occurs whenever a new full node is created; the node must obtain (usually through downloading) most of or all the blockchain data (Bitcoin's blockchain data is over 175 gigabytes and growing as of this writing) [30]. This process uses a lot of network bandwidth.

7.9 Inadequate Block Publishing Rewards

A potential limitation is the risk of inadequate rewards for publishing a block. The combination of increased competition, increased computational resources needed to have meaningful contributions to pools of publishing nodes, and highly volatile market prices in the cryptocurrency market creates the risk that the expected return for any given cryptocurrency may be less than the power costs needed to run publishing node software. Thus, the expected return for other cryptocurrencies may be more attractive.

Cryptocurrencies that are not able to consistently and adequately reward publishing nodes risk delays in publishing blocks and processing transactions. These delays could therefore reduce confidence in the cryptocurrency, reducing its market value further. It could then become increasingly less attractive for publishing nodes to contribute to that cryptocurrency's publishing efforts. Even worse, such weakened cryptocurrencies open themselves up to being attacked by nodes with large amounts of resources that may maliciously alter the blockchain or deny service to users attempting to submit transactions.

7.10 Public Key Infrastructure and Identity

When hearing that blockchain technology incorporates a public key infrastructure, some people immediately believe it intrinsically supports identity. This is not the case, as there may not be a one-to-one relationship of private key pairs to users (a user can have multiple private keys), nor is there a one-to-one relationship between blockchain addresses and public keys (multiple addresses can be derived from a single public key).

Digital signatures are often used to prove identity in the cybersecurity world, and this can lead to confusion about the potential application of a blockchain to identity management. A blockchain's transaction signature verification process links transactions to the owners of private keys but provides no facility for associating real-world identities with these owners. In some cases, it is possible to connect real-world identities with private keys, but these connections are made through processes outside, and not explicitly supported by, the blockchain. For example, a law enforcement agency could request records from an exchange that would connect transactions to specific individuals. Another example is an individual posting a cryptocurrency address on their personal website or social media page for donations, this would provide a link from address to real world identity.

While it is possible to use blockchain technology in identity management frameworks that require a distributed ledger component, it is important to understand that typical blockchain implementations are not designed to serve as standalone identity management systems. There is more to having secure digital identities than simply implementing a blockchain.

8 Application Considerations

Since blockchain technology is still new, a lot of organizations are looking at ways to incorporate it into their businesses. The fear of missing out on this technology is quite high, and most organizations approach the problem as "we want to use blockchain somewhere, where can we do that?" which leads to frustrations with the technology as it cannot be applied universally. A better approach would be to first understand blockchain technology, where it fits, and then identify systems (new and old) that may fit the blockchain paradigm.

Blockchain technology solutions may be suitable if the activities or systems require features such as:

- Many participants
- Distributed participants
- Want or need for lack of trusted third party
- Workflow is transactional in nature (e.g., transfer of digital assets/information between parties)
- A need for a globally scarce digital identifier (i.e., digital art, digital land, digital property)
- A need for a decentralized naming service or ordered registry
- A need for a cryptographically secure system of ownership
- A need to reduce or eliminate manual efforts of reconciliation and dispute resolutions
- A need to enable real time monitoring of activity between regulators and regulated entities
- A need for full provenance of digital assets and a full transactional history to be shared amongst participants

Several agencies and organizations have developed guides to help determine if a blockchain is suitable for a particular system or activity, and which kind of blockchain technology would be of most benefit. In this section, some articles and advice are highlighted from several different sectors – federal government, academia, technical publications, technology websites, and software developers.

The United States Department of Homeland Security (DHS) Science & Technology Directorate has been investigating blockchain technology and has created a flowchart to help one determine whether a blockchain may be needed for a development initiative. The flowchart is reproduced here, with permission.

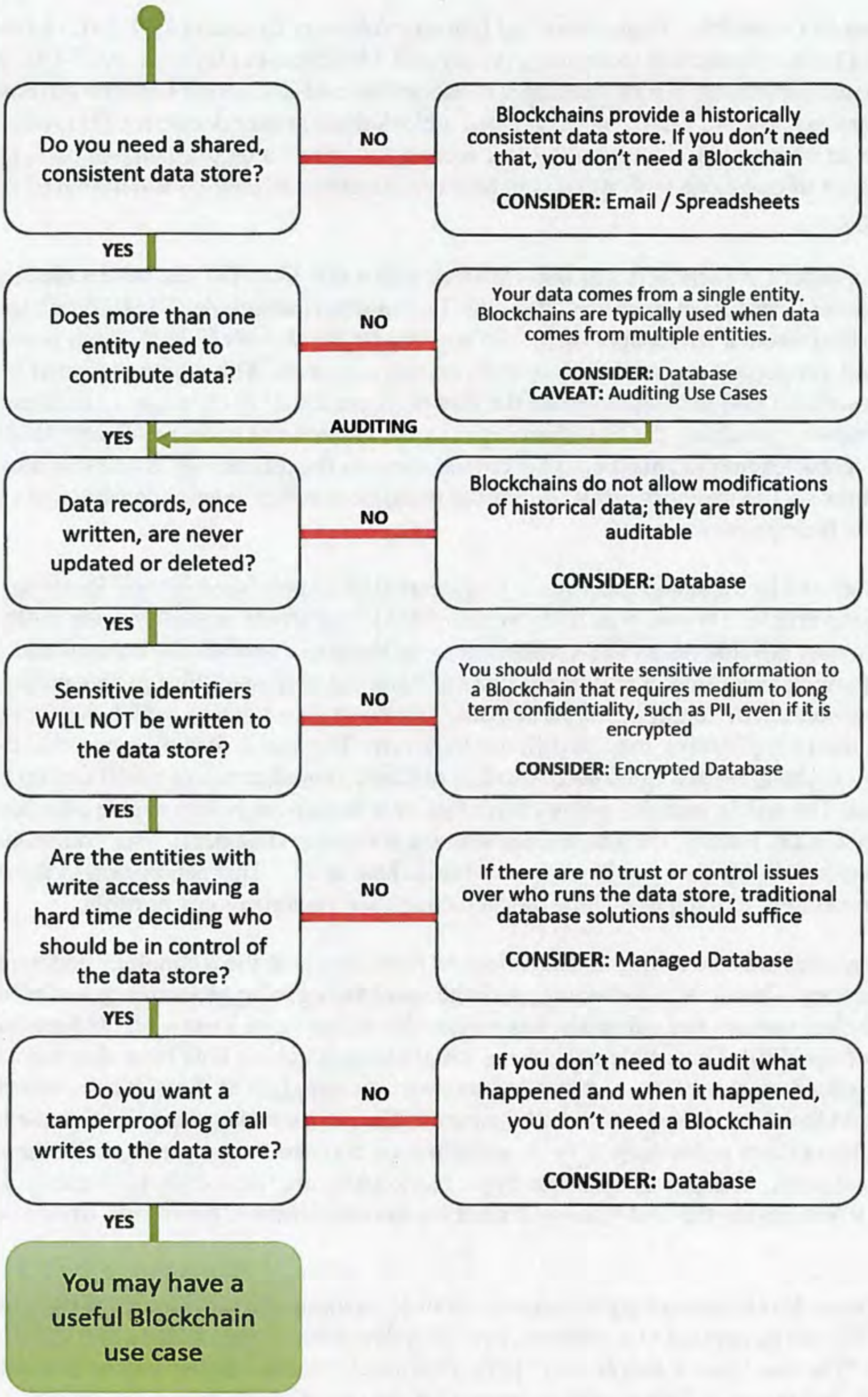

Do you need a shared, consistent data store?
NO
Blockchains provide a historically consistent data store. If you don't need that, you don't need a Blockchain
CONSIDER: Email / Spreadsheets
YES
Does more than one entity need to contribute data?
NO
Your data comes from a single entity. Blockchains are typically used when data comes from multiple entities.
CONSIDER: Database
CAVEAT: Auditing Use Cases
AUDITING
YES
Data records, once written, are never updated or deleted?
NO
Blockchains do not allow modifications of historical data; they are strongly auditable
CONSIDER: Database
YES
Sensitive identifiers WILL NOT be written to the data store?
NO
You should not write sensitive information to a Blockchain that requires medium to long term confidentiality, such as PII, even if it is encrypted
CONSIDER: Encrypted Database
YES
Are the entities with write access having a hard time deciding who should be in control of the data store?
NO
If there are no trust or control issues over who runs the data store, traditional database solutions should suffice
CONSIDER: Managed Database
YES
Do you want a tamperproof log of all writes to the data store?
NO
If you don't need to audit what happened and when it happened, you don't need a Blockchain
CONSIDER: Database
YES
You may have a useful Blockchain use case

Figure 6 - DHS Science & Technology Directorate Flowchart

The American Council for Technology and Industry Advisory Council (ACT-IAC) have been developing both a blockchain technology primer and a blockchain playbook. ACT-IAC is a public/private partnership which facilitates collaboration and discussion between government and industry experts. ACT-IAC has developed a blockchain primer document [31], which aims to provide an overview of the technology. A second document, a blockchain playbook [32], provides a set of questions with weights to help organizations in their consideration of the technology.

There is no lack of whitepapers and news articles with a title like "Do you need a blockchain?" Two computer scientists at the Eidgenössische Technische Hochschule (ETH) Zürich university in Switzerland wrote a whitepaper titled "*Do you need a Blockchain?*" [33] which provides the background, properties, and a critical view on several use cases. Although not created by the authors, a website [34] has implemented the flowchart presented in the paper in an interactive form. However, examining the flowchart logic, as well as website code, most paths lead to "no" with only a few leading to "maybe." This critical view on the technology is one that most organizations should take; organizations should examine whether existing technologies can better solve their problems.

The Institute of Electrical and Electronics Engineers (IEEE) published in their *Spectrum* magazine the article "*Do you need a blockchain?*"[35]. The article emphasizes the utility a blockchain may provide (as an anti-censorship tool), but also discusses the tradeoff that must be made by moving away from a traditional system. Removal of trusted third parties means relying on multiple sources of "unaffiliated participants" acting in coordination, which depending on the type of blockchain platform, may be difficult to govern. The article also discusses that the technology is changing at a rapid pace – so it is difficult to predict where it will end up in a few years' time. The article includes a flowchart of its own to help the reader decide whether they need a blockchain. Finally, the article ends with the following statement: "But you should also consider the possibility that you don't need a blockchain at all." This is pertinent to those who may be desperately looking to include blockchain in their organization's portfolio.

Technology sites are also asking organizations to look closely at the technology and apply it only when necessary. Coindesk, a technology website specializing in cryptocurrency and blockchain news, technical matters and editorials, has written the article "*Don't use a blockchain unless you really need one*"[36]. The article gives some small examples about how most data today is owned by siloed organizations, and that as users we only supply it to them. It asks what the world would look like if users owned all their data. The article makes the point that the largest benefit of blockchain technology is its decentralization and can be summed up with the article's most critical point: "Despite some of the hype, blockchains are 'incredibly inefficient,' Ravikant said. 'It's worth paying the cost when you need the decentralization, but it's not when you don't.'"

Even software developers are urging organizations to examine the key aspects of the technology and how it could be applied to a problem. One such developer wrote on the website C# Corner the article "*Do You Need A Blockchain*" [37]. This article touches on the history of blockchain technology and brings to light a primary reason for the use of blockchain technology: "Blockchain brings trust to a transactional system."

By utilizing a blockchain cryptographic trust can be introduced into a previously no to low trust system. The article goes on to ask several pointed questions (and provides a flowchart) for helping to decide whether a blockchain network would be of benefit.

While several sources have been mentioned above for deciding if a blockchain would be applicable, there are many more. Most of the advice surrounding blockchain technology is: investigate it and use it if it is appropriate – not because it is new.

8.1 Additional Blockchain Considerations

When deciding whether to utilize a blockchain, one must take into consideration additional factors and determine if these factors limit one's ability to use a blockchain or a particular type of blockchain:

- **Data Visibility**
  - Permissioned blockchain networks may or may not reveal blockchain data publicly. The data may only be available to those within the blockchain network. Consider scenarios where data may be governed by policy or regulations (such as Personally Identifiable Information (PII) or General Data Protection Regulation (GDPR) regulations). Data such as this may or may not be appropriate to store even within a permissioned blockchain network.
  - Permissionless blockchain networks can allow anyone to inspect and contribute to the blockchain. The data is generally public. This leads to several questions that must be considered. Does the data for the application need to be available to everyone? Is there any harm to having public data?
- **Full transactional history** – Some blockchain networks provide a full public history of a digital asset – from creation, to every transaction it is included in. This feature may be beneficial for some solutions, and not beneficial for others.
- **Fake Data Input** – Since multiple users are contributing to a blockchain, some could submit false data, mimicking data from valid sources (such as sensor data). It is difficult to automate the verification of data that enters a blockchain network. Smart contract implementations may provide additional checks to help validate data where possible.
- **Tamper evident and tamper resistant data** – Many applications follow the "CRUD" (create, read, update, delete) functions for data. With a blockchain, there is only "CR" (create, read). There are methods that can be employed to "deprecate" older data if a newer version is found, but there is no removal process for the original data. By using new transactions to amend and update previous transactions, data can be updated while providing a full history. However, even if a new transaction marked an older transaction as "deleted" – the data would still be present in the blockchain data, even if it is not shown within an application processing the data.
- **Transactions Per Second** – Transaction processing speed is highly dependent on the consensus model used. Currently transactions on many permissionless blockchain networks are not executed at the same pace as other information technology solutions due to a slow publication time for blocks (usually in terms of seconds, but sometimes minutes). Thus, some slowdown in blockchain dependent applications may occur while

waiting for data to be posted. One must ask if their application can handle relatively slow transaction processing?

- **Compliance** – The use of blockchain technology does not exclude a system from following any applicable laws and regulations. For example, there are many compliance considerations with regards to legislation and policies tied to PII or GDPR that identify that certain information should not be placed on the blockchain. In addition, certain countries may limit the type of data that can be transferred across its geographic boundary. In other instances, certain legislation may dictate that the "first write" of financial transactions must be written to a node which is present within their borders. In any of these cases, a public, permissionless chain may be less appropriate, with a permissioned or hybrid approach required to satisfy regulatory needs.
  An additional example of laws and regulations are for any blockchain network which manages federal records. Federal records are subject to many laws and regulations.[11] Federal agencies themselves must follow specific federal guidelines when utilizing blockchain technology.[12]
- **Permissions** – For permissioned blockchain networks, there are considerations around the permissions themselves
  - Granularity – do the permissions within the system allow for enough granularity for specific roles that users may need (in a manner like Role-Based Access Control methods) to perform actions within the system
    - Permissioned blockchain networks allow for more traditional roles such as administrator, user, validator, auditor, etc.
  - Administration – who can administer permissions? Once permissions are administered to a user, can they easily be revoked?
- **Node Diversity** – A blockchain network is only as strong as the aggregate of all the existing nodes participating in the network. If all the nodes share similar hardware, software, geographic location, and messaging schema then there exists a certain amount of risk associated with the possibility of undiscovered security vulnerabilities. This risk is mitigated through the decentralization of the network of heterogeneous devices, which may be defined as "the non-shared characteristics between any one node and the generalized set"

[11] Such as found in the National Archives and Records Administration handbook https://www.archives.gov/records-mgmt/handbook/records-mgmt-language.html

[12] Such as found in the National Archives and Administration policy guide https://www.archives.gov/records-mgmt/policy/universalermrequirements

9 Conclusions

Blockchain technology is a new tool with potential applications for organizations, enabling secure transactions without the need for a central authority. Starting in 2009[13], with Bitcoin leveraging blockchain technology, there has been an increasing number of blockchain technology-based solutions.

The first applications were electronic cash systems with the distribution of a global ledger containing all transactions. These transactions are secured with cryptographic hashes, and transactions are signed and verified using asymmetric-key pairs. The transaction history efficiently and securely records a chain of events in a way that any attempt to edit or change a past transaction will also require a recalculation of all subsequent blocks of transactions.

The use of blockchain technology is still in its early stages, but it is built on widely understood and sound cryptographic principles. Currently, there is a lot of hype around the technology, and many proposed uses for it. Moving forward, it is likely that the hype will die down, and blockchain technology will become just another tool that can be used.

As detailed throughout this publication, a blockchain relies on existing network, cryptographic, and recordkeeping technologies but uses them in a new manner. It will be important that organizations are able to look at the technologies and both the advantages and disadvantages of using them. Once a blockchain is implemented and widely adopted, it may become difficult to change it. Once data is recorded in a blockchain, that data is usually there forever, even when there is a mistake. Applications that utilize the blockchain as a data layer work around the fact that the actual blockchain data cannot be altered by making later blocks and transactions act as updates or modifications to earlier blocks and transactions. This software abstraction allows for modifications to working data, while providing a full history of changes. For some organizations these are desirable features. For others, these may be deal breakers preventing the adoption of blockchain technology.

Blockchain technology is still new and organizations should treat blockchain technology like they would any other technological solution at their disposal--use it only in appropriate situations.

[13] Although the whitepaper *Bitcoin: A Peer-to-Peer Electronic Cash System* was published in 2008, the actual Bitcoin network would not launch until 2009.

Appendix A—Acronyms

Selected acronyms and abbreviations used in this paper are defined below.

ACM	Association for Computing Machinery
ACT-IAC	American Council for Technology and Industry Advisory Council
ASIC	Application-Specific Integrated Circuit
BCH	Bitcoin Cash
BFT	Byzantine Fault Tolerant
BTC	Bitcoin
CPU	Central Processing Unit
CR	Create, Read
CRUD	Create, Read, Update, Delete
DAG	Directed Acyclic Graph
DAO	Decentralized Autonomous Organization
DHS	Department of Homeland Security
DID	Decentralized Identifier
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ETC	Ethereum Classic
ETH	Ethereum
EVM	Ethereum Virtual Machine
FIPS	Federal Information Processing Standard
FOIA	Freedom of Information Act
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
I2P	Invisible Internet Project

IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IR	Internal Report
ITL	Information Technology Laboratory
KYC	Know Your Customer
LDAP	Lightweight Directory Access Protocol
NARA	National Archives and Records Administration
NIST	National Institute of Standards and Technology
NISTIR	National Institute of Standards and Technology Internal Report
MB	Megabyte
PII	Personally Identifiable Information
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
QR	Quick Response
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RSA	Rivest-Shamir-Adleman
SegWit	Segregated Witness
SHA	Secure Hash Algorithm
XRP	Ripple

Appendix B—Glossary

Selected terms used in this paper are defined below.

Address
A short, alphanumeric string derived from a user's public key using a hash function, with additional data to detect errors. Addresses are used to send and receive digital assets.

Assets
Anything that can be transferred.

Asymmetric-key cryptography
A cryptographic system where users have a private key that is kept secret and used to generate a public key (which is freely provided to others). Users can digitally sign data with their private key and the resulting signature can be verified by anyone using the corresponding public key.

Also known as Public-key cryptography.

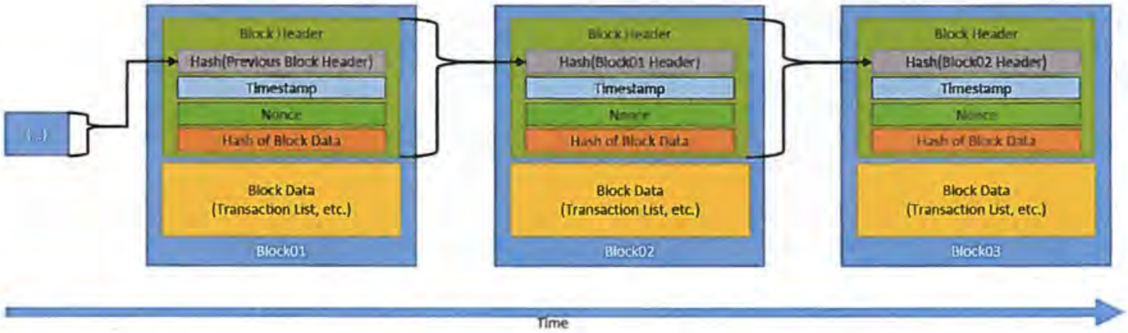
Block
A data structure containing a block header and block data.

Block data
The portion of a block that contains a set of validated transactions and ledger events.

Block header
The portion of a block that contains information about the block itself (block metadata), typically including a timestamp, a hash representation of the block data, the hash of the previous block's header, and a cryptographic nonce (if needed).

Block reward
A reward (typically cryptocurrency) awarded to publishing nodes for successfully adding a block to the blockchain.

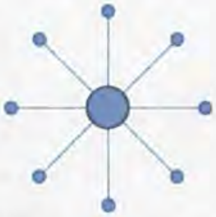
Blockchain
Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance). New blocks are replicated across copies of the ledger within the network, and any conflicts are resolved automatically using established rules.



Blockchain network user
Any single person, group, business, or organization which is using or operating a blockchain node.

Byzantine fault tolerant proof of stake consensus model
A proof of stake consensus model where the blockchain decides the next block by allowing all staked members to “vote” on which submitted block to include next.

Centralized network
A network configuration where participants must communicate with a central authority to communicate with one another. Since all participants must go through a single centralized source, the loss of that source would prevent all participants from communicating.



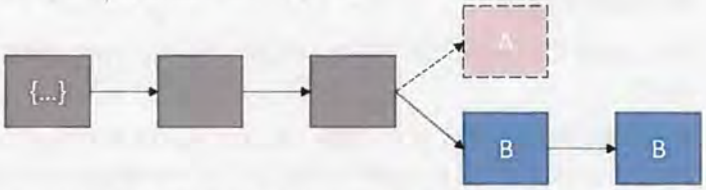
Chain-based proof of stake consensus model
A proof of stake consensus model where the blockchain network decides the next block through pseudo-random selection, based on a personal stake to overall system asset ratio.

Checksum
Value computed on data to detect error or manipulation.

Confirmed
State of a transaction or block when consensus has been reached about its status of inclusion into the blockchain.

Conflict
One or more participants disagree on the state of the system.

Conflict resolution
A predefined method for coming to a consensus on the state of the system. For example, when portions of the system participants claim there is `State_A` and the rest of the participants claim there is `State_B`, there is a conflict. The system will automatically resolve this conflict by choosing the “valid” state as being the one from whichever group adds the next block of data. Any transactions “lost” by the state not chosen are added back into the pending transaction pool.



Consensus model
A process to achieve agreement within a distributed system on the valid state.

Also known as a *consensus algorithm, consensus mechanism, consensus method.*

Cryptocurrency
A digital asset/credit/unit within the system, which is cryptographically sent from one blockchain network user to another. In the case of cryptocurrency creation (such as the reward for mining), the publishing node includes a transaction sending the newly created cryptocurrency to one or more blockchain network users.

These assets are transferred from one user to another by using digital signatures with asymmetric-key pairs.

Cryptographic hash function

A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties:

1. (*Preimage resistant*) It is computationally infeasible to compute the correct input value given some output value (the hash function is 'one way').
2. (*Second preimage resistant*) One cannot find an input that hashes to a specific output.
3. (*Collision resistant*) It is computationally infeasible to find any two distinct inputs that map to the same output.

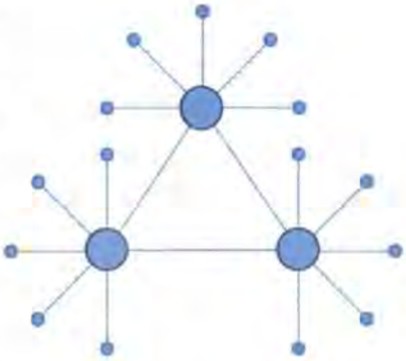
See the NIST SP 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms, http://dx.doi.org/10.6028/NIST.SP.800-175B.

Cryptographic nonce

An arbitrary number that is used once.

Decentralized network

A network configuration where there are multiple authorities that serve as a centralized hub for a subsection of participants. Since some participants are behind a centralized hub, the loss of that hub will prevent those participants from communicating.



Digest

See hash digest

Digital asset

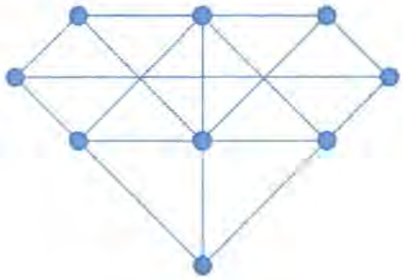
Any asset that is purely digital, or is a digital representation of a physical asset

Digital signature

A cryptographic technique that utilizes asymmetric-keys to determine authenticity (i.e., users can verify that the message was signed with a private key corresponding to the specified public key), non-repudiation (a user cannot deny having sent a message) and integrity (that the message was not altered during transmission).

Distributed network

A network configuration where every participant can communicate with one another without going through a centralized point. Since there are multiple pathways for communication, the loss of any participant will not prevent communication.



This is also known as a peer-to-peer network.

Double spend (problem) — Transacting with the same set of digital assets more than once. This is a problem which has plagued many digital money systems, and a problem that most blockchain networks are designed to prevent.

Double spend (attack) — An attack where a blockchain network user attempts to explicitly double spend a digital asset.

Fault tolerance — A property of a system that allows proper operation even if components fail.

Fork — A change to blockchain network's software (usually the consensus algorithm). The changes may be backwards compatible - see Soft Fork, or the changes may not be backwards compatible - see Hard Fork.

Full node — A blockchain node that stores the blockchain data, passes along the data to other nodes, and ensures that newly added blocks are valid and authentic.

Genesis block — The first block of a blockchain network; it records the initial state of the system.

Hard fork — A change to a blockchain implementation that is not backwards compatible. Non-updated nodes cannot continue to transact with updated nodes.

Hash chain — An append-only data structure where data is bundled into data blocks that include a hash of the previous data block's data within the newest data block. This data structure provides evidence of tampering because any modification to a data block will change the hash digest recorded by the following data block.

Hash digest — The output of a hash function (e.g., hash(data) = digest).

Also known as a message digest, digest or hash value.

Hash rate — The number of cryptographic hash functions a processor can calculate in a given time, usually denominated as hashes per second.

Hash value — See Hash digest.

Hashing — A method of calculating a relatively unique output (called a *hash digest*) for an input of nearly any size (a file, text, image, etc.) by applying a cryptographic hash function to the input data.

Immutable — Data that can only be written, not modified or deleted.

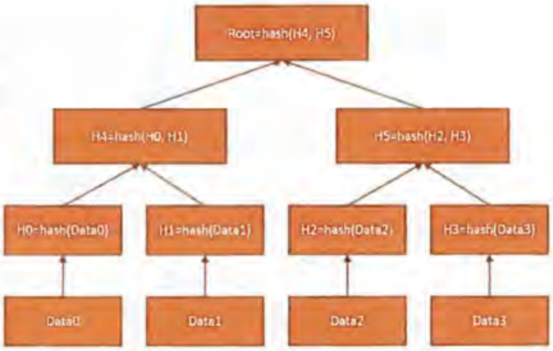
Incentive mechanism — See Reward system

Ledger — A record of transactions.

Lightweight node — A blockchain node that does not need to store a full copy of the blockchain and often passes its data to full nodes to be processed.

Merkle tree

A data structure where the data is hashed and combined until there is a singular root hash that represents the entire structure.



Mining

The act of solving a puzzle within a proof of work consensus model.

Pending transaction pool

A distributed queue where candidate transactions wait until they are added to the blockchain.

Also known as memory pool, or mempool.

Publishing node

A node that, in addition to all responsibilities required of a full node, is tasked with extending the blockchain by creating and publishing new blocks.

Also known as a mining node, committing node, minting node.

Node

An individual system within the blockchain network.

Nonce

See Cryptographic Nonce

Orphan block

Any block that is not in the main chain after a temporary ledger conflict.

Permissioned

A system where every node, and every user must be granted permissions to utilize the system (generally assigned by an administrator or consortium).

Permissionless

A system where all users' permissions are equal and not set by any administrator or consortium.

Permissions

Allowable user actions (e.g., read, write, execute).

Proof of stake consensus model

A consensus model where the blockchain network is secured by users locking an amount of cryptocurrency into the blockchain network, a process called *staking*. Participants with more stake in the system are more likely to want it to succeed and to not be subverted, which gives them more weight during consensus.

Proof of work consensus model

A consensus model where a publishing node wins the right to publish the next block by expending time, energy, and computational cycles to solve a hard-to-solve, but easy-to-verify problem (e.g., finding the nonce which, when combined with the data to be added to the block, will result in a specific output pattern).

Public key cryptography

See Asymmetric-key cryptography.

Reward system
A means of providing blockchain network users an award for activities within the blockchain network (typically used as a system to reward successful publishing of blocks).

Also known as incentive system.

Round robin consensus model
A consensus model for permissioned blockchain networks where nodes are pseudo-randomly selected to create blocks, but a node must wait several block-creation cycles before being chosen again to add another new block. This model ensures that no one participant creates the majority of the blocks, and it benefits from a straightforward approach, lacking cryptographic puzzles, and having low power requirements.

Smart contract
A collection of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the blockchain network. The smart contract is executed by nodes within the blockchain network; all nodes must derive the same results for the execution, and the results of execution are recorded on the blockchain.

Soft fork
A change to a blockchain implementation that is backwards compatible. Non-updated nodes can continue to transact with updated nodes.

Tamper evident
A process which makes alterations to the data easily detectable.

Tamper resistant
A process which makes alterations to the data difficult (hard to perform), costly (expensive to perform), or both.

Transaction
A recording of an event, such as the transfer of assets (digital currency, units of inventory, etc.) between parties, or the creation of new assets.

Transaction fee
An amount of cryptocurrency charged to process a blockchain transaction. Given to publishing nodes to include the transaction within a block.

Turing complete
A system (computer system, programming language, etc.) that can be used for any algorithm, regardless of complexity, to find a solution.

Wallet
Software used to store and manage asymmetric-keys and addresses used for transactions.

Appendix C—References

[1] Clarke, A.C., "Hazards of Prophecy: The Failure of Imagination," from *Profiles of the Future: An Inquiry into the Limits of the Possible*, 1962.

[2] Lamport, Leslie. "The Part-Time Parliament." *ACM Transactions on Computer Systems*, vol. 16, no. 2, Jan. 1998, pp. 133–169., https://dl.acm.org/citation.cfm?doid=279227.279229.

[3] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfede, S., *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press, 2016.

[4] Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. https://bitcoin.org/bitcoin.pdf

[5] National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, Federal Information Processing Standards (FIPS) Publication 180-4, August 2015. https://doi.org/10.6028/NIST.FIPS.180-4

[6] National Institute of Standards and Technology (NIST), Secure Hashing website, https://csrc.nist.gov/projects/hash-functions

[7] "Hash per Second." *Bitcoin Wiki*, http://en.bitcoin.it/wiki/Hash_per_second.

[8] National Institute of Standards and Technology, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Federal Information Processing Standards (FIPS) Publication 202, August 2015. https://doi.org/10.6028/NIST.FIPS.202

[9] National Institute of Standards and Technology (NIST), *Digital Signature Standard*, Federal Information Processing Standards (FIPS) Publication 186-4, July 2013. https://doi.org/10.6028/NIST.FIPS.186-4

[10] "LDAP.com." *LDAP.com*, https://www.ldap.com.

[11] "How Is the Address of an Ethereum Contract Computed?" *Ethereum Stack Exchange*, 29 Jan. 2016, 22:14, https://ethereum.stackexchange.com/questions/760/how-is-the-address-of-an-ethereum-contract-computed.

[12] Bahsoun, J.P., Guerraoui, R., and Shoker, A., "Making BFT Protocols Really Adaptive," *2015 IEEE International Parallel and Distributed Processing Symposium*, Hyderabad, India, pp. 904-913, 2015. https://doi.org/10.1109/IPDPS.2015.21

[13] Lamport, L. "Time, Clocks, and the Ordering of Events in a Distributed System." Communications of the ACM, vol. 21, no. 7, January 1978, pp. 558–565., doi:10.1145/359545.359563. https://amturing.acm.org/p558-lamport.pdf.

[14] Todd, P. Bitcoin Improvement Protocol (BIP) 65, "OP_CHECKLOCKTIMEVERIFY," October 1, 2014. https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki

[15] Wong, J. and Kar, I., "Everything you need to know about the Ethereum 'hard fork,'" *Quartz Media*, July 18, 2016. https://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/

[16] Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D., *Report on Post-Quantum Cryptography*, National Institute of Standards and Technology Internal Report (NISTIR) 8105, April 2016. https://doi.org/10.6028/NIST.IR.8105

[17] Szabo, N. "Smart Contracts," 1994. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[18] Mell, P., Kelsey, J., and Shook, J., "Cryptocurrency Smart Contracts for Distributed Consensus of Public Randomness." October 7, 2017. https://doi.org/10.1007/978-3-319-69084-1_31

[19] "Majority Attack." *Bitcoin Wiki*, https://en.bitcoin.it/wiki/Majority_attack.

[20] Greenspan, G. "The Blockchain Immutability Myth." CoinDesk, May 9, 2017, https://www.coindesk.com/blockchain-immutability-myth/.

[21] Narayanan, A., "Analyzing the 2013 Bitcoin fork: centralized decision-making saved the day," MultiChain, July 28, 2015. https://freedom-to-tinker.com/2015/07/28/analyzing-the-2013-bitcoin-fork-centralized-decision-making-saved-the-day

[22] Buck, J. "Blockchain Oracles, Explained." Cointelegraph, October 18, 2017, https://cointelegraph.com/explained/blockchain-oracles-explained

[23] https://medium.com/@kleffew/truthpoint-angelhacks-dc-submission-5569252d795a

[24] Greenspan, G., "The Blockchain Immutability Myth," MultiChain, May 4, 2017. https://www.multichain.com/blog/2017/05/blockchain-immutability-myth/

[25] de Vries, A. "Bitcoin's Growing Energy Problem." *Joule*, vol. 2, no. 5, 16 May 2018, pp. 801–805., https://doi.org/10.1016/j.joule.2018.04.016.

[26] Deetman, S., "Bitcoin Could Consume as Much Electricity as Denmark by 2020," *Motherboard*, March 29, 2016. https://motherboard.vice.com/en_us/article/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020

[27] Hern, A., "Bitcoin mining consumes more electricity a year than Ireland," *The Guardian*, November 27, 2017. https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland

[28] Power Compare, https://powercompare.co.uk/bitcoin/

[29] Loh, T. "Bitcoin's Power Needs May Be Overblown." *Bloomberg.com*, Bloomberg, January 16, 2018, https://www.bloomberg.com/news/articles/2018-01-16/bitcoin-s-power-needs-may-be-overblown-recalling-pot-growing.

[30] "Blockchain Size." Blockchain.com, www.blockchain.com/charts/blocks-size. Accessed July 19, 2018.

[31] "ACT-IAC White Paper: Enabling Blockchain Innovation in the U.S. Federal Government." ACT-IAC, February 27, 2018, https://www.actiac.org/act-iac-white-paper-enabling-blockchain-innovation-us-federal-government.

[32] "ACT-IAC White Paper: Blockchain Playbook for the U.S. Federal Government." ACT-IAC, April 23, 2018, https://www.actiac.org/act-iac-white-paper-blockchain-playbook-us-federal-government.

[33] Wüst, K., Gervais, A. "Do You Need a Blockchain?" IACR ePrint Archive, 2017, p. 375., https://eprint.iacr.org/2017/375.pdf.

[34] "Do You Need a Blockchain?" Do You Need a Blockchain?, http://doyouneedablockchain.com/.

[35] Peck, Morgen E. "Do You Need a Blockchain?" IEEE Spectrum: Technology, Engineering, and Science News, IEEE Spectrum, September 29, 2017, https://spectrum.ieee.org/computing/networks/do-you-need-a-blockchain.

[36] Hochstein, M. "Don't Use a Blockchain Unless You Really Need One." CoinDesk, CoinDesk, 16 Jan. 2018, https://www.coindesk.com/dont-use-blockchain-unless-really-need-one/.

[37] Chand, M. "Do You Need A Blockchain." C# Corner, January 4, 2018, https://www.c-sharpcorner.com/article/do-you-need-a-blockchain2/.

EXHIBIT 139

How to Become a Relayer?

web.archive.org/web/20220609181519/https://docs.tornado.cash/general/how-to-become-a-relayer

Following the execution of Tornado Cash 10th governance proposal, anyone can become a relayer for Tornado Cash users.
The only condition to be included on the Tornado Cash UI is to lock a min. of `300 TORN` *. To remain listed, it is needed to keep enough TORN locked (~ `40 TORN` at the moment in April 2022) to be able to pay back the transaction fee to the staking contract.

**This minimum stake can be changed by a governance vote at any time.*

Relayers form an essential & necessary part of the Tornado Cash ecosystem. Their use guarantees privacy as they solve the infamous “fee payment dilemma? : how to pay fees for token withdrawals from a pool while maintaining anonymity?

Therefore, relayers act as third parties and manage the entire withdrawal. They pay for transaction fees by deducting them directly from the transferred amount. They also charge an additional fee for their services.

Since the implementation of the Relayer Registry proposal, the protocol collects a fee directly from the relayer’s staked balance through the `StakingReward` contract for each withdrawal. This fee percentage may vary from one pool to another and is also subject to change through on-chain governance.
Currently, it is fixed at `0.3%` . Some pools remain without fees, either because the instance is too small to assign a fee (0.1 ETH, 100 DAI/USDT, 1000 DAI/USDT), or because there is not enough liquidity on Uni v3 (all cDAI instances).

Anyone can become a relayer for the protocol in **6 simple steps** through a Relayer Registry User Interface (UI).
Below you will find everything your need to join our relayers' club & get listed on Tornado Cash decentralized relayer registry.

---

1. Warning: Understand & Accept Potential Risks

Before you commit to sharing part of your journey with Tornado Cash users as a relayer, you need to understand & accept all potential risks of being a relayer for the protocol.

How a Relayer is chosen by user interface

The formula for designating a relayer is as follows:

The list of all registered relayers is retrieved from the Relayer Registry smart contract.

For each relayer, calculate a score based on its staked TORN and its fee. The higher the stake, the higher the score is; the higher the fee, the lower the score is. For Ethereum mainnet, the formula used to calculate the score is `stake * [1 - 25*(fee-0.33)^2]` ; for sidechains, the formula is `stake * [1 - 11.89*(fee-0.01)^2]` .

Then randomly pick a relayer, weighted by its calculated score.

---

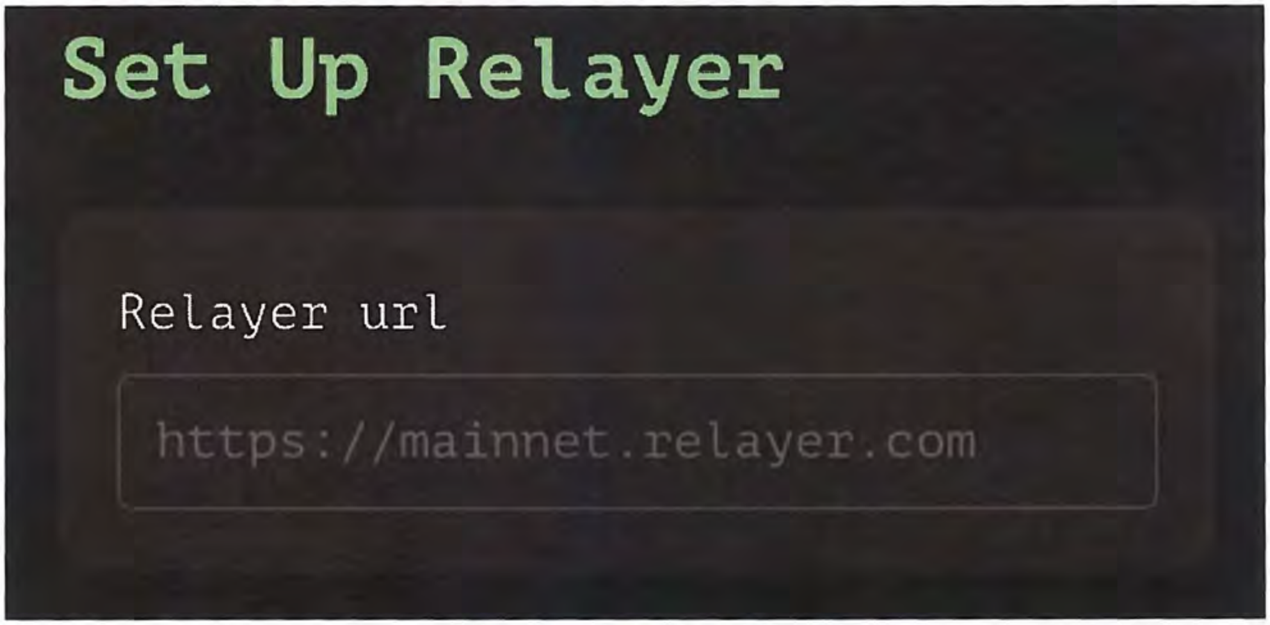
2. Set up Relayer

The first concrete step is to run the Tornado Cash Relayer software for Ethereum Mainnet on your computer. All steps are outlined in the protocol's github. To complete this task successfully, you will have to carefully follow these instructions.

GitHub - tornadocash/tornado-relayer: Relayer for Tornado cash.

GitHub

Once completed, you will need to insert your url in the input box.



Set Up Relayer

Relayer url

https://mainnet.relayer.com

It is strongly recommended that you use your own RPC nodes. Instructions on how to run full nodes can be found here.

---

3. Set Up ENS Subdomain

The next steps entail:

Creating an ENS domain for your relayer.

Setting up its mainnet subdomain.

Adding a TXT record with the Relayer URL to the mainnet subdomain according to this specific format:

Ethereum Relayers (Mandatory)

TXT record

mainnet-tornado.xxx.eth

goerli-tornado.xxx.eth

Sidechains Relayers (Optional)

You also have the option to add subdomains with their corresponding TXT records to support chains other than Ethereum. Sidechains relayers use a different version of the Relayer software. The complete requirements with instructions are found here.
TXT record

bsc-tornado.xxx.eth

gnosis-tornado.xxx.eth

polygon-tornado.xxx.eth

optimism-tornado.xxx.eth

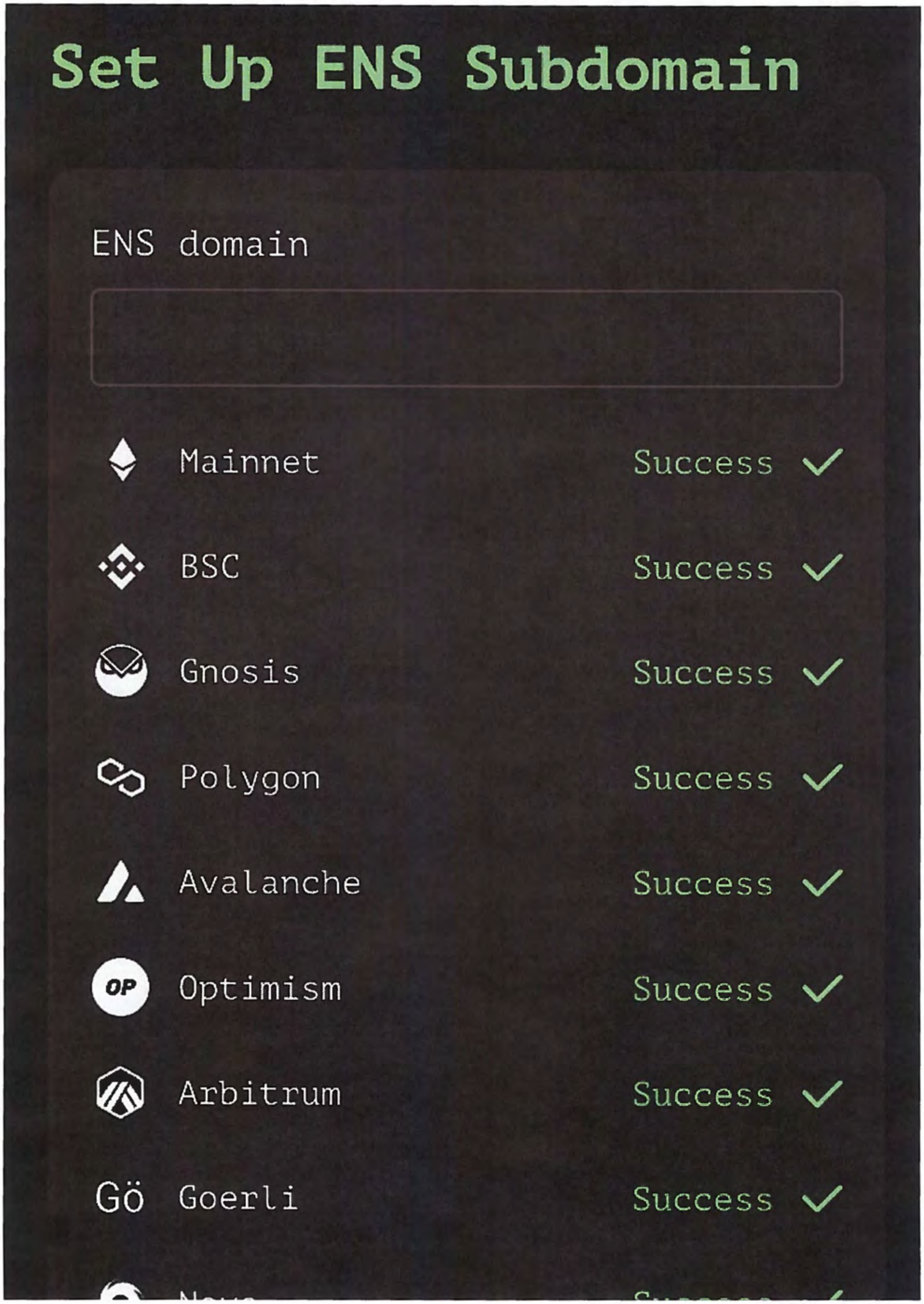
arbitrum-tornado.xxx.eth

avalanche-tornado.xxx.eth

Nova Relayer (Optional)

Tornado Cash Nova uses its own version of the software. If you wish to become a relayer for Tornado Cash Nova, you will find instructions to follow here.
TXT record

gnosis-nova.xxx.eth



Set Up ENS Subdomain

ENS domain

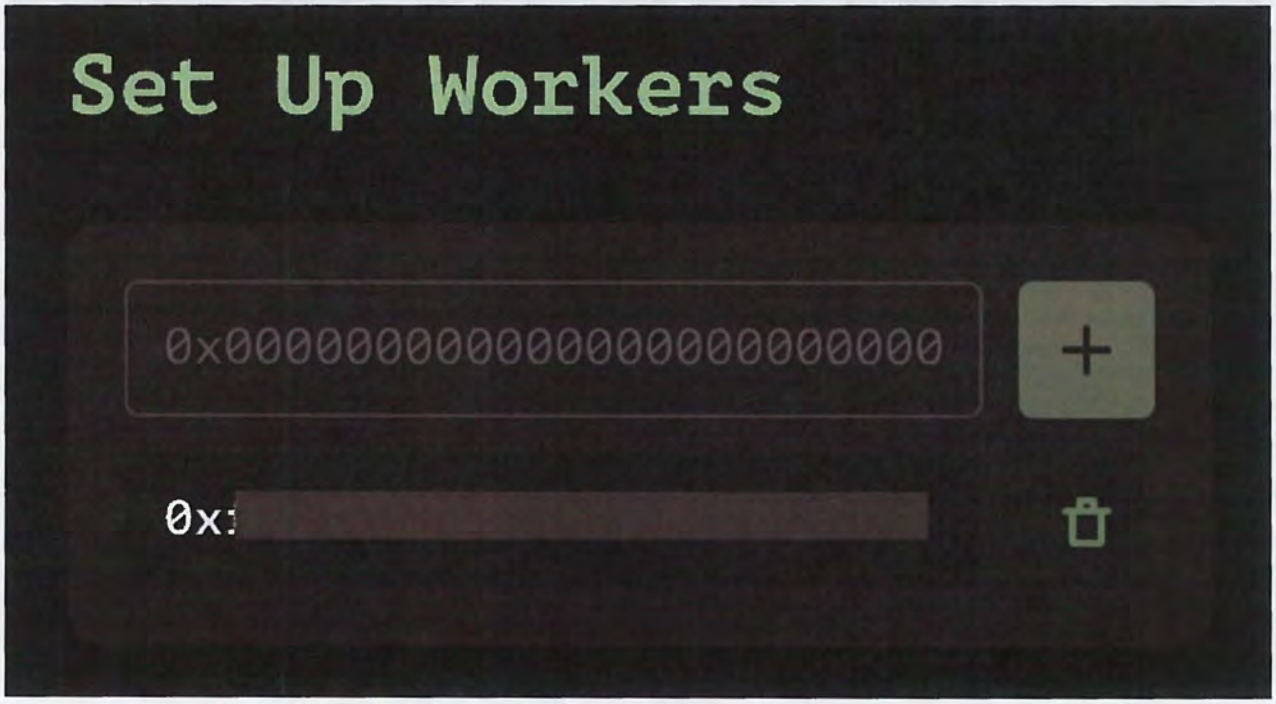
Mainnet	Success ✓
BSC	Success ✓
Gnosis	Success ✓
Polygon	Success ✓
Avalanche	Success ✓
Optimism	Success ✓
Arbitrum	Success ✓
Gö Goerli	Success ✓
Nova	Success ✓

4. Set Up Workers

Workers are the addresses that will allow your relayer to send ZK-proofs to users. By default, the first worker is the ENS domain owner's address.

To ensure an extra level of security, we advise you to set up more than one worker.

Only the mainnet requires you to register workers. All other networks do not require the use of registered workers.



Set Up Workers

0x000000000000000000000000 +

0x: [redacted]

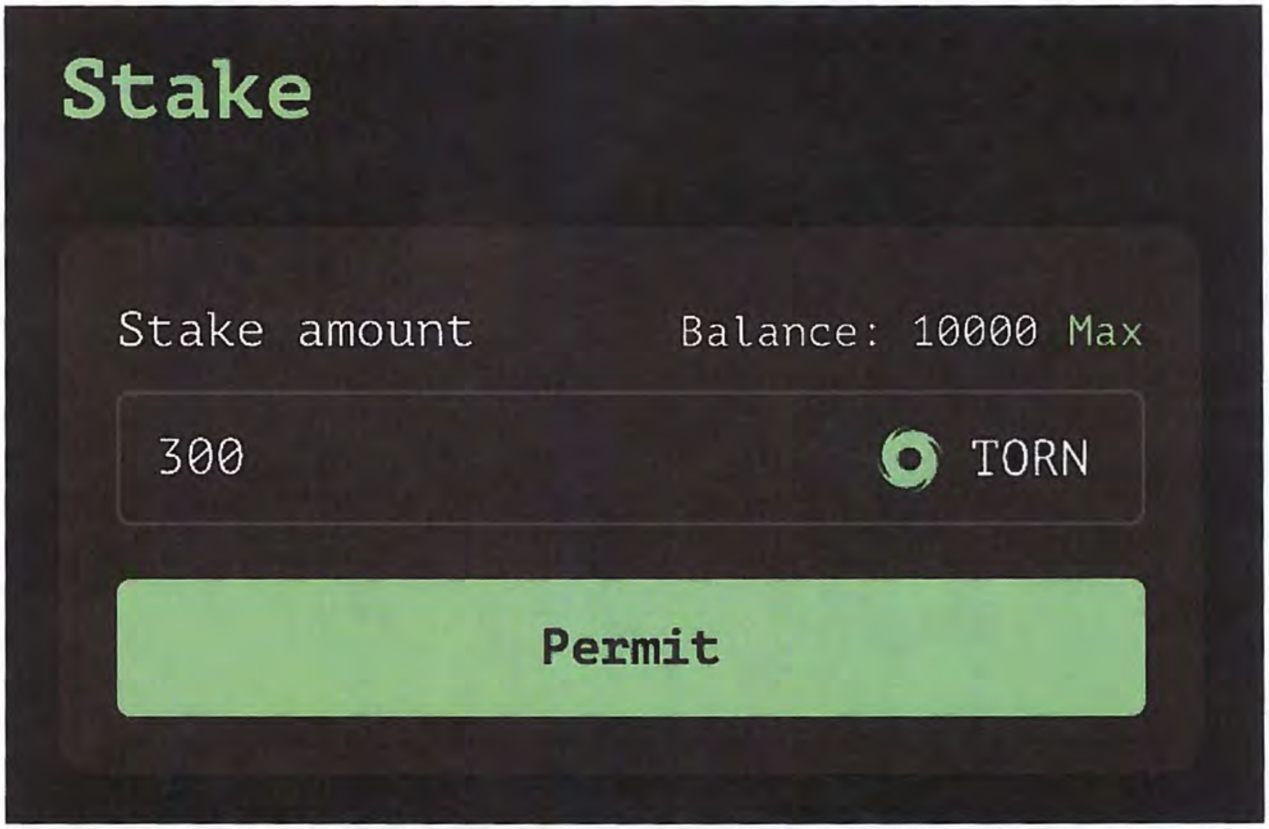
5. Stake

With the implementation of a decentralized relayer registry, a staking condition has been introduced as a requirement to become listed on Tornado Cash UI. Keep in mind **staking TORN is now necessary to be added to the recommended list of relayers.**

The minimum staked amount is currently set by Tornado Cash governance at `300 TORN` . This threshold can be changed by Tornado Cash governance at any time.

When a relayer is used in the Tornado Cash pool, a small amount of TORN is automatically collected from this staked balance by the `StakingReward` contract. This element is essential to keep in mind as relayers will need to keep enough TORN locked (~ `40 TORN` at the moment in April 2022) to be able to pay back the transaction fee to the staking contract.

The collected fees are subsequently distributed among DAO members with locked TORN tokens. TORN are usually locked to participate in on-chain governance (submitting & voting on proposals). You can find more information both on this forum post & in the Staking TORN documentation page.
Your staked TORN amount is not claimable, and it is non-refundable.



Stake

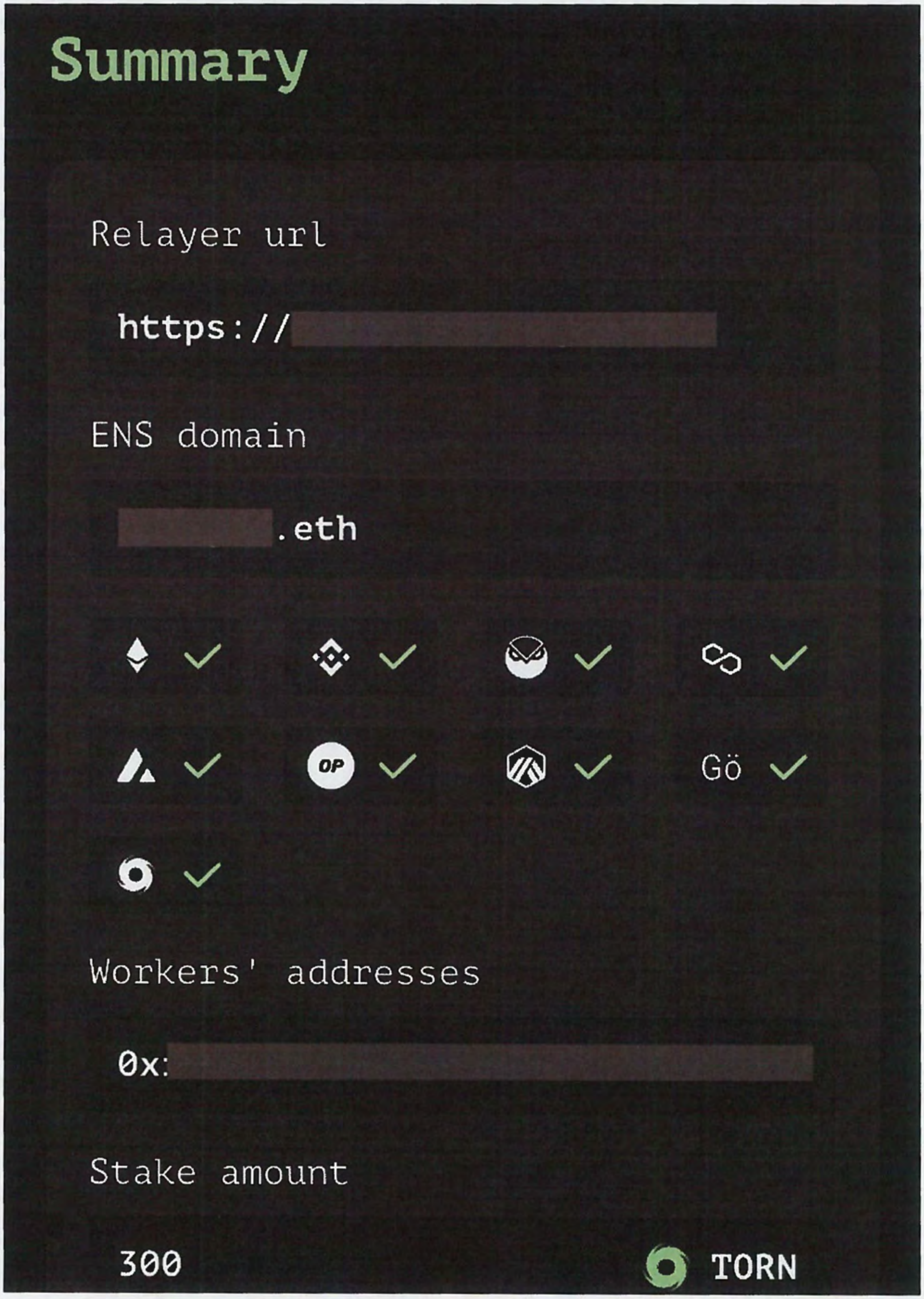
Stake amount Balance: 10000 Max

300 TORN

Permit

6. Summary: Final Verification & Registration

Last but not least, we advise you to **double-check all information** displayed in the Summary before registering.



Summary

Relayer url

https://[redacted]

ENS domain

[redacted].eth

✓ ✓ ✓ ✓

✓ ✓ ✓ Gö ✓

✓

Workers' addresses

0x:[redacted]

Stake amount

300 TORN

[redacted]

Welcome to the relayer team! Thanks to you, privacy can be better respected 🩶

Written by ***@bt11ba*** *&* ***@ayefda***

EXHIBIT 157

[redacted]

Decentralized autonomous organizations (DAOs)

- Member-owned communities without centralized leadership.
- A safe way to collaborate with internet strangers.

- lace to commit funds to a specific cause.

On this page ›

≡ Ethereum use cases

DAOs are an effective and safe way to work with like-minded folks around the globe.

Think of them like an internet-native business that's collectively owned and managed by its members. They have built-in treasuries that no one has the authority to access without the approval of the group. Decisions are governed by proposals and voting to ensure everyone in the organization has a voice.

There's no CEO who can authorize spending based on their own whims and no chance of a dodgy CFO manipulating the books. Everything is out in the open and the rules around spending are baked into the DAO via its code.

Why do we need DAOs?

Starting an organization with someone that involves funding and money requires a lot of trust in the people you're working with. But it's hard to trust someone you've only ever interacted with on the internet. With DAOs you don't need to trust anyone else in the group, just the DAO's code, which is 100% transparent and verifiable by anyone.

This opens up so many new opportunities for global collaboration and coordination.

p so many new opportunities for global collaboration and coordination.

A comparison

DAO	A traditional organization
Usually flat, and fully democratized.	Usually hierarchical.
Voting required by members for any changes to be implemented.	Depending on structure, changes can be demanded from a sole party, or voting may be offered.
Votes tallied, and outcome implemented automatically without trusted intermediary.	If voting allowed, votes are tallied internally, and outcome of voting must be handled manually.
Services offered are handled automatically in a decentralized manner (for example distribution of philanthropic funds).	Requires human handling, or centrally controlled automation, prone to manipulation.
All activity is transparent and fully public.	Activity is typically private, and limited to the public.

DAO examples

To help this make more sense, here's a few examples of how you could use a DAO:

- A charity – you can accept membership and donations from anyone in the world and the group can decide how they want to spend donations.
- A freelancer network – you could create a network of contractors who pool their funds for office spaces and software

subscriptions.

- Ventures and grants – you could create a venture fund that pools investment capital and votes on ventures to back. Repaid money could later be redistributed amongst DAO-members.

DAO membership

There are different models for DAO membership. Membership can determine how voting works and other key parts of the DAO.

Token-based membership

Usually fully permissionless, depending on the token used. Mostly these governance tokens can be traded permissionlessly on a decentralized exchange. Others must be earned through providing liquidity or some other 'proof-of-work'. Either way, simply holding the token grants access to voting.

Typically used to govern broad decentralized protocols and/or tokens themselves.

A famous example

MakerDAO ↗ – MakerDAO's token MKR is widely available on decentralized exchanges. So anyone can buy into having voting power on the Maker protocol's future.

Share-based membership

Share-based DAOs are more permissioned, but still quite open. Any prospective members can submit a proposal to join the DAO, usually offering a tribute of some value in the form of tokens or work. Shares represent direct voting power and ownership. Members can exit at any time with their proportionate share of the treasury.

Typically used for more closer-knit, human-centric organizations like charities, worker collectives, and investment clubs. Can also govern protocols and tokens as well.

A famous example

MolochDAO ↗ – MolochDAO is focused on funding Ethereum projects. They require a proposal for membership so the group can assess whether you have the necessary expertise and capital to make informed judgments about potential grantees. You can't just buy access to the DAO on the open market.

How do DAOs work?

The backbone of a DAO is its smart contract. The contract defines the rules of the organization and holds the group's treasury. Once the contract is live on Ethereum, no one can change the rules except by a vote. If anyone tries to do something that's not covered by the rules and logic in the code, it will fail. And because the treasury is defined by the smart contract too that means no one can spend the money without the group's approval either. This means that DAOs don't need a central authority. Instead, the group makes decisions collectively, and payments are automatically authorized when votes pass.

This is possible because smart contracts are tamper-proof once they go live on Ethereum. You can't just edit the code (the DAOs rules) without people noticing because everything is public

[redacted]

out people noticing because everything is public.

More on smart contracts →

Ethereum and DAOs

Ethereum is the perfect foundation for DAOs for a number of reasons:

- Ethereum's own consensus is distributed and established enough for organizations to trust the network.
- Smart contract code can't be modified once live, even by its owners. This allows the DAO to run by the rules it was programmed with.
- Smart contracts can send/receive funds. Without this you'd need a trusted intermediary to manage group funds.
- The Ethereum community has proven to be more collaborative than competitive, allowing for best practices and support systems to emerge quickly.

Join / start a DAO

Join a DAO

- Ethereum community DAOs
- DAOHaus's list of DAOs ↗

Start a DAO

- Summon a DAO with DAOHaus ↗
- Create an Aragon-powered DAO ↗
- Start a colony ↗
- Build a DAO with DAOstack ↗

Further reading

DAO Articles

- What's a DAO? ↗ – Aragon ↗
- House of DAOs ↗ – Metagame ↗
- What is a DAO and what is it for? ↗ – DAOhaus ↗
- How to Start a DAO-Powered Digital Community ↗ – DAOhaus ↗
- What is a DAO? ↗ – Coinmarketcap ↗

Videos

- What is a DAO in crypto? ↗

[redacted]

Website last updated: April 25, 2022

Use Ethereum

Ethereum wallets

Get ETH

Decentralized applications (dapps)

Layer 2

Run a node

Stablecoins

Stake ETH

Learn

What is Ethereum?

What is ether (ETH)?

Community guides and resources

History of Ethereum

Ethereum Whitepaper

Ethereum upgrades

Ethereum security and scam prevention

Ethereum glossary ?

Ethereum governance

Blockchain bridges

[redacted]

Ethereum energy consumption

What is Web3?

Ethereum Improvement Proposals

Developers

Get started

Documentation

Tutorials

Learn by coding

Set up local environment

Ecosystem

Community hub

Ethereum Foundation

Ethereum Foundation Blog ↗

Ecosystem Support Program ↗

Ecosystem Grant Programs

Ethereum brand assets

Devcon ↗

Enterprise

Mainnet Ethereum

Private Ethereum

Enterprise

About ethereum.org

About us

Jobs

Contributing

Language support

Privacy policy

Terms of use

Cookie policy

Contact ↗